Provable Instantiations of Correlation Intractability and the Fiat-Shamir Heuristic

by

Alex Lombardi

A.B., Harvard University (2016)
A.M., Harvard University (2016)
S.M., Massachusetts Institute of Technology (2018)

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2022

Author . . . .
Department of Electrical Engineering and Computer Science
August 26, 2022

Certified by . . . .
Vinod Vaikuntanathan
Professor of Electrical Engineering and Computer Science
Thesis Supervisor

Accepted by . . . .
Leslie A. Kolodziejski
Professor of Electrical Engineering and Computer Science
Chair, Department Committee on Graduate Students

Provable Instantiations of Correlation Intractability and the Fiat-Shamir Heuristic

by

Alex Lombardi

Submitted to the Department of Electrical Engineering and Computer Science
on August 26, 2022, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Abstract

Interactive proof systems, introduced in a seminal work of Goldwasser, Micali, and Rackoff, have become one of the most powerful and flexible tools in cryptography and computer science at large. They have directly led to some of the biggest breakthroughs in theoretical cryptography, complexity, and quantum computation. They are also at the center of a revolution in practical cryptography, particularly in the context of blockchains and cryptocurrencies.

However, despite their importance, our understanding of cryptographic proofs is surprisingly limited. The central problem studied in this thesis is the following question:

Can we remove interaction from interactive proofs?

Even though this question sounds almost paradoxical, Fiat and Shamir (1986) proposed (and Blum extended) a heuristic methodology for removing interaction from a huge class of interactive proofs. This methodology is ubiquitous and essential for practical applications, but for over thirty years, we had no proof of its security, even for a single non-trivial case.

The main goal of this thesis is to give a solid theoretical foundation for the Fiat-Shamir transformation by developing general-purpose tools, techniques, and abstractions for characterizing its security. We propose a two-step methodology for obtaining provable instantiations that relies on the notion of correlation intractability, which is a hash function security property requiring that it is computationally infeasible to find pre-specified input-output correlations in the hash function.

Using this methodology, we obtain various new results in cryptography, touching on areas such as non-interactive zero knowledge, delegation of computation, the insecurity of parallel repetition, and the cryptographic hardness of computing Nash Equilibria in game theory.

Thesis Supervisor: Vinod Vaikuntanathan
Title: Professor of Electrical Engineering and Computer Science

Acknowledgments

I was extremely fortunate to spend my graduate school years surrounded by wonderful people – collaborators, colleagues, friends, and family – that shared with me their drive, curiosity, inspiration, and kindness. Without them, this thesis would not have been possible.

My advisor, Vinod Vaikuntanathan, was incredibly generous with his time, attention, patience, and thoughts. Vinod constantly and enthusiastically discussed with me countless open problems and research directions that excited him. I specifically credit this for developing my high-level understanding of cryptography as a whole: what we know, what we would like to know, what ideas are developing at the moment, and how everything fits together. He also deliberately and actively introduced me to the cryptography community and encouraged outside collaborations that proved to be highly impactful and influential. My conception of what a great researcher is and does largely stems from Vinod's explicit advice and his leading by example. At the same time, Vinod gave me the freedom to be my own researcher and was a great source of personal support and understanding throughout my PhD. I could not have asked for a better advisor.

The work done in this thesis was in joint collaboration with many amazing researchers: Ran Canetti, Yilei Chen, Justin Holmgren, Fermi Ma, Willy Quach, Guy Rothblum, Ron Rothblum, Vinod Vaikuntanathan, and Daniel Wichs. I thank them for helping to make this research possible and, by virtue of the collaborations, helping me develop as a cryptographer and as a person. I would also like to thank my collaborators in works not appearing in this thesis: Prabhanjan Ananth, James Bartusek, Zvika Brakerski, Yael Kalai, Giulio Malavolta, Luke Schaeffer, Gil Segev, Nick Spooner, Thomas Vidick, June Vuong, David Wu, and Lisa Yang.

I would specifically like to thank a number of research mentors – Zvika Brakerski, Ran Canetti, Justin Holmgren, Yael Kalai, Ron Rothblum, and Daniel Wichs – for, at various points in my career, taking me on as a "charge" and going far out of their way to share their knowledge and learn alongside me. Each of them has had a lasting impact on how I think about both research and mentorship, and they are all inspiration for any future collaboration I might have with younger people and students especially. I am especially indebted to Justin Holmgren, who managed to do all of these things *while a student himself* (at the beginning), and whom I collaborated with closely for much of the work in this thesis.

It takes a village to raise a graduate student, and the MIT theory group was an extremely vibrant and welcoming village to grow up in. At the beginning of graduate school, I knew quite little about TCS and cryptography; fortunately, I had an amazing time learning by osmosis from all of the students, postdocs, and faculty around me. I specifically want to thank all of the cryptography students and postdocs who were at MIT when I started: Itay Berman, Nir Bitansky, Aloni Cohen, Ran Cohen, Akshay Degwekar, Justin Holmgren, Rio LaVigne, Tianren Liu, Omer Paneth, Ron Rothblum, Sunoo Park, Srinivasan Raghuraman, Adam Sealfon, and Prashant Vasudevan. I was also fortunate to have a collaborative and supportive cohort who joined MIT at the same time as me, including Willow Ahrens, Sitan Chen, Robin Hui, Quanquan Liu, Aleksandar Makelov, Saleet Mossel, Nicholas Schiefer, and Helen Xu.

I thank Michael Cohen for being uniquely willing to (enthusiastically) talk about math/TCS for an arbitrary amount of time, especially late at night on the 5th floor of Stata; Michael will be remembered fondly by everyone who knew him.

Finally, I want to thank all of the theory group friends who joined me in various shenanigans over the years including theory retreat, cards, crosswords, chess, Pokemon Go, Super Smash Bros, etc. The full list of these people would be incredibly long, but let me specifically thank Aviv Adler, Michael Coulombe, Daniel Grier, Dhiraj Holden, Gautam Kamath, Pritish Kamath, Jerry Li, Andrea Lincoln, Dylan McKay, Saeed Mehraban, Madalina Persu, Govind Ramnarayan, Luke Schaeffer, Mike Sun, Nicole Wein, and Kai Xiao, in addition to everyone I already mentioned.

Beyond the MIT theory group, I was constantly energized and supported by friends who were with me through good and bad times. Life was enjoyable and worthwhile because I had all of them around. I want to thank all of my friends from college, especially Keno Fischer, Patrick Komiske, Eric Metodiev, and Erik Tamre, who all remained in Cambridge with me during grad school (Keno and Patrick as roommates, who were there for me when I needed it the most). Fermi Ma and Willy Quach became close friends during grad school; I greatly enjoyed all of our intellectual conversations, climbing and restaurant adventures, video games, and summer of living together. Andy Lu, Eric Lu, and Yuna Joung have been entertaining, crazy, and incredibly kind roommates for the last couple of years.

Brynmor Chapman went through all of the last five-and-a-half years with me, including a large fraction of everything mentioned so far. I have no idea how to thank him properly, so we are going to have to settle for "you know what you did."

Finally, I want to thank my family: my parents (Robin Lewis and Alfred Lombardi), my brother (Nick Lombardi), and my partner (Jess Xu). Their unconditional love and support gives me the strength to be my best self, and I would not have made it without them. Jess, the last three years have been the best of my life so far, and I can't wait for what comes next.

Contents

Chapter 1

Introduction

One of computer science's greatest insights has been in understanding the power and versatility of *proofs*. Nowhere is this more clearly demonstrated than in a seminal work of Goldwasser, Micali, and Rackoff [GMR85], which established a radically new conception of proofs that are *interactive* and *randomized*. In such a proof system, one party (P, "the prover") wants to convince another (V, "the verifier") of the validity of some statement x. In order to do so, the prover and verifier exchange messages according to a pre-specified protocol; at the end, the verifier decides whether to accept the prover's claim that x is true.

It is hard to overstate how powerful and versatile this computational model has been over the last thirty-five years. The model has an extraordinary tendency to identify fundamental questions about and insights into the nature of computation. This stems (in part) from how it simultaneously incorporates three individually powerful computational resources: **randomization** (the ability of an algorithm to flip coins), **interaction** (the ability to exchange many back-and-forth messages before the verifier makes a decision), and **computational hardness** (the conjectured *inability* of efficient algorithms to solve important computational tasks). Computational hardness can appear in multiple aspects of a proof system, but one instance worth immediately highlighting is *computational soundness*: it is possible to construct proof systems where valid proofs of false statements *exist* but it is computationally infeasible to find one.

Since their introduction, interactive proofs have directly led to some of the biggest breakthroughs in cryptography [GMW86, GMW87, Kil92, Mic94] as well as related areas such as complexity [LFKN90, Sha90, BFLS91, ALM+92] and, more recently, quantum computation [BCM+18, Mah18]. We elaborate on some examples below:

1. [GMR85] introduced and constructed *zero knowledge* interactive proof systems, which are proof systems in which the verifier learns *nothing* beyond the validity of the statement x. This property is formalized by a simulation security definition requiring that the entire "view" of any efficient (potentially malicious) verifier V^* can be simulated in polynomial time given the statement x. We note that security is typically only required to hold against polynomial-time V^*.

   In addition to the amazing standalone result that any NP language has a zero-knowledge proof system [GMW86], zero-knowledge proofs have become a ubiquitous tool in cryptography starting from their use in constructing general-purpose secure multiparty computation [GMW87].

2. Lund, Fortnow, Karloff, Nisan, and Shamir [LFKN90, Sha90] demonstrated the incredible power of interactive proofs by proving that IPs with a polynomial-time verifier and computationally unbounded prover exist for all languages in PSPACE (polynomial space). These techniques were later "scaled down" to construct similarly powerful interactive proofs for polynomial-time computation [GKR08, RRR16] (with super-efficient verifiers) and also strongly influenced complexity-theoretic results about the related computational model of "probabilistically checkable proofs" (PCPs) [BFLS91, FGL+91, ALM+92].

3. Kilian and Micali [Kil92, Mic94] introduced and constructed *succinct, computationally sound* interactive proof systems (hereafter called succinct arguments). In these proof systems, the prover and verifier exchange an *extremely short* (polynomial in a security parameter λ) transcript that enables the verifier to check the validity of the statement x *more efficiently* than would have been possible without the prover. For example, they showed that *every* NP language

has a succinct argument system where the verifier runs in time $\mathsf{poly}(\lambda, |x|)$, *independent* of the size of the NP witness or NP language's verification time.

Succinct and zero-knowledge arguments are also now at the center of a revolution in *practical* cryptography [ZKP], particularly in the context of blockchains and cryptocurrencies [W+14, SCG+14, CM19, Sta].

Removing Interaction. Despite the fact that they have been studied intensely for thirty-five years, our understanding of cryptographic proofs is extremely limited. In this thesis, we focus primarily on the following basic question:

Question 1.1. *Can we* remove interaction *from interactive proof systems?*

In many of the most compelling applications of cryptographic proof systems, it is highly desirable to use protocols that are **non-interactive**. Non-interactive proofs can be written down, transferred from person to person, and be verified by *anyone*, a feature that is often important in the digital world. However, Question 1.1 may sound paradoxical or nonsensical: what was the point of introducing the interactive proofs model if it was possible to remove the interaction anyway?

There are a couple of ways to address this confusion. First of all, as will be explained below, there are necessarily going to be trade-offs when interaction is removed; specifically, proof systems will be made non-interactive using a *hash function* family, and the security of this transformation must rely on a security property of the hash function (even if the original proof system had no cryptography in it). So in some sense, Question 1.1 can be viewed as asking about *trading* interaction for additional computational hardness assumptions.

Second of all, even if it may have been possible to design a non-interactive proof system from the start, it has proved extremely fruitful – both in understanding the feasibility and in obtaining concretely efficient constructions – to solve cryptographic tasks by first designing an interactive protocol and then removing interaction with a compiler. Indeed, this thesis concerns the *Fiat-Shamir heuristic* [FS87], a general-purpose compiler for this task.

1.1 The Fiat-Shamir Heuristic

Question 1.1 (in our cryptographic context) was first studied in the 1980s by Fiat, Shamir, and Blum[1] [FS87, BR93]. Specifically, they introduced a highly influential transformation that *generically* removes interaction from a wide class of interactive protocols. The transformation can be in principle applied to any interactive protocol that is *public-coin*, meaning that (1) all verifier messages in the protocol are uniformly random strings, and (2) the verifier's decision is a public, deterministic function of the protocol transcript. While certainly a non-trivial condition, public-coin protocols are quite common and also can sometimes be obtained generically [GS86].

Given any such protocol Π *and a hash function family* $\mathcal{H}$, the Fiat-Shamir heuristic compiles Π into a two-message protocol $\Pi_{\text{FS}} = \Pi_{\text{FS},\mathcal{H}}$, as follows.

- The Π_{FS} verifier first sends a description of a hash function h.
- The Π_{FS} prover responds with the transcript of an emulated execution of Π (including an input x, as well as all messages exchanged between the prover and verifier), in which each verifier message is set to be the value of h applied to the transcript so far.
- The Π_{FS} verifier checks that the transcript it received is consistent with h, and that the verifier of Π would have accepted.

The resulting protocol Π_{FS} from this transformation satisfies many highly desirable properties: it is non-interactive (h could be chosen ahead of time as part of a common reference string), it is publicly verifiable, and the communication and computational overhead of Π_{FS} is minimal over that of Π!

In practice, the Fiat-Shamir transform has been heuristically used as the basis for many important protocols, including identification and signature schemes, succinct non-interactive arguments (SNARGs) and non-interactive zero-knowledge protocols (NIZKs), e.g. [FS87, BR93, Mic94, PS96, BCS16, WTs+18]. The Fiat-Shamir transform

[1] [BR93] credits Blum (via personal communication through Micali and Rudich) for formulating the general-purpose variant of the transformation introduced in [FS87].

was (and remains) indispensable for these constructions largely due to (1) its simplicity (it adds little in computational overhead or complexity) and (2) its generality (it can be applied to a huge class of protocols).

The specific form of Question 1.1 studied in this thesis is understanding the security of this transformation:

Question 1.2. *For which protocols and hash families does the Fiat-Shamir transform preserve soundness? Under what assumptions can we prove this?*

Unfortunately, and despite its importance, the Fiat-Shamir transformation was largely known to have only heuristic [FS87, BR93, CGH98] and poorly understood [Bar01, GK03, GW11] security. Its main justification appeals to the random oracle model [BR93]: If h is modeled as a random oracle (to which the adversary only has query-access), then Π_{FS} is sound as long as Π is computationally sound and either has a constant number of rounds [FS87, PS96, AABN02] or more generally, satisfies a stronger soundness property called *soundness against state restoration attacks* [BCS16].

Are we done? While we have already declared the security of the Fiat-Shamir transform to be "poorly understood," let us pause to discuss why this is the case:

- **Efficiently computable random oracles do not exist.** First and foremost, while the random oracle heuristic has been indispensable for constructing and justifying hash function-based cryptographic schemes, it has been known for decades [CGH98] that security in the random oracle model does *not* imply that secure instantiations exist. This is especially relevant in our context: there are constructions of protocols [Bar01, GK03], to which Fiat-Shamir can be securely applied in the ROM, such that applying Fiat-Shamir with *any* efficiently computable hash function family results in a broken protocol.

- **Hash functions in the real world are not used as oracles.** In addition to the fact that ROM security does not imply standard model security for our protocols, the model itself simply does *not* accurately represent how hash functions are used in the real world. For example, it is very common for proof systems to be *recursively composed* [Val08, BCCT13] so that proofs can be *bootstrapped* (from simpler to more complex statements) as well as *updated* over time. Security using concrete hash function families is essential in order for the security of such composition to be established.

- **The line between "possible" and "impossible" is not understood.** Given impossibility results such as [Bar01, GK03], one could hope to identify a way to *distinguish* between protocols with sound Fiat-Shamir instantiations and those without. Unfortunately, we do not currently know how to do this. For example, it is even unclear how strong of an impossibility result holds for Kilian's highly influential protocol [BBH+19].

- **We want provable security.** Given these serious issues with the random oracle model, how is security typically argued *without* heuristics? Whenever possible, the theory of cryptography follows a framework pioneered in [GM84]:

  - Formulate a strong and comprehensive *definition* of what it means for a cryptosystem to be secure, and

  - Give a *security reduction* showing that violating the definition would imply an efficient algorithm for a *simple, well-studied* computational problem believed to be intractable.

  The Fiat-Shamir heuristic has thus far resisted analysis within this (standard) framework. For example, Bitansky et al. [BDSG+13] show that, even restricting to three-round (statistically sound) *proofs* (avoiding the above impossibility results), soundness of the general-purpose Fiat-Shamir transform with a concrete hash family cannot be proved via black box reduction to a standard "falsifiable" assumption [Nao03, GW11].

Finally, perhaps best illustrating our poor understanding, we note that for over thirty years (until this thesis), we had no proof of security for the Fiat-Shamir transformation applied to **even a single non-trivial proof system** based on a standard computational assumption.

1.2 Cryptographic Hash Functions and Correlation Intractability

Understanding the security of the Fiat-Shamir heuristic is primarily a question about *hash functions*: what kind of hash function should be used, and how can security when using this hash function be proven?

This question is subtle because we do not have a general-purpose definition of a secure hash function. In the context of constructing digital signatures and other specific applications, security properties such as **one-wayness** [DH76, Mer79], **universal one-wayness** [Mer79, NY89] and **collision resistance** [Mer79, Dam88] were initially proposed and used. However, there is no single security definition capturing all common hash function use cases. The formalism that comes the closest to this goal is the random oracle model [BR93], but as discussed above, the ROM is uninstantiable in the real world.

Given the above state of affairs, this thesis also studies the following basic question about hash functions:

Question 1.3. *Which random oracle properties can be instantiated with* concrete hash functions*? Under what computational assumptions?*

We study Question 1.3 from the perspective of **correlation intractability** [CGH98], a simple-to-state but extremely expressive family of hash function security properties.

Definition 1.4 (Correlation Intractability, informal)**.** *A hash function family $\mathcal{H}$ is* correlation intractable *for a $2t$-ary relation $R(x_1, \ldots, x_t, y_1, \ldots, y_t)$ if the following problem is computationally hard: given a hash function $h \leftarrow \mathcal{H}$, find t inputs $x_1, \ldots, x_t$ such that $(x_1, \ldots, x_t, h(x_1), \ldots, h(x_t)) \in R$.*

For some choices of relation R (such as the trivial R containing every string), this property is clearly unsatisfiable (even if you had a random oracle). On the other hand, for all R satisfying an appropriate notion of *sparsity*, it is not hard to show that a random oracle is R-correlation intractable. Moreover, a few special cases of correlation intractability are quite familiar to the cryptography community:

- When $t = 1$ and $R = \{(x, 0)\}$, R-correlation intractability corresponds to the hardness of inverting the hash function h at 0, a form of one-wayness.
- When $t = 2$ and $R = \{(x_1, x_2, y_1, y_2) : x_1 \neq x_2 \text{ and } y_1 = y_2\}$, R-correlation intractability is exactly collision-resistance.
- It is folklore knowledge [DNRS99] that general-purpose correlation intractability in the $t = 1$ case ("single-input CI") has important connections to Question 1.2.

Thus, the more specific form of Question 1.3 studied in this thesis is as follows.

Question 1.5. *Can we build correlation-intractable hash functions? Which kinds, and under which computational assumptions?*

1.3 Results

The main goal of this thesis is to give a solid theoretical foundation for correlation intractability and the Fiat-Shamir heuristic. We will do so by developing **general-purpose tools, techniques, and abstractions** for characterizing the security of these objects. Finally, we will *apply* these ideas and tools to obtain various new feasibility results in cryptography.

Our results appear in the following papers (listed in chronological order):

- Cryptographic Hashing from Strong One-Way Functions (or: One-Way Product Functions and their Applications), by Justin Holmgren and Alex Lombardi [HL18].
- Fiat-Shamir, From Practice to Theory, by Ran Canetti, Yilei Chen, Justin Holmgren, Alex Lombardi, Guy N. Rothblum, and Ron D. Rothblum, and Daniel Wichs [CCH$^+$19].

  - Part I: Fiat-Shamir From Simpler Assumptions, by Ran Canetti, Yilei Chen, Justin Holmgren, Alex Lombardi, Guy N. Rothblum, and Ron D. Rothblum [CCH+18].
  - Part II: Non-Interactive Zero Knowledge and Correlation Intractability from Circular-Secure FHE, by Ran Canetti, Alex Lombardi, and Daniel Wichs [CLW18].

- 2-Message Publicly Verifiable WI from (Subexponential) LWE, by Alex Lombardi, Vinod Vaikuntanathan, and Daniel Wichs [LVW19].

- Fiat-Shamir for Repeated Squaring and Applications to PPAD-Hardness and VDFs, by Alex Lombardi and Vinod Vaikuntanathan [LV20a].

- Does Fiat-Shamir Require a Cryptographic Hash Function? by Yilei Chen, Alex Lombardi, Fermi Ma, and Willy Quach [CLMQ21]. We only include one auxiliary result (Theorem A.1) from [CLMQ21] that is most relevant to this thesis.

- Correlation-Intractable Hash Functions via Shift Hiding, by Alex Lombardi and Vinod Vaikuntanathan [LV20b].

- Fiat-Shamir via List-Recoverable Codes (or: Parallel Repetition of GMW is Not Zero Knowledge), by Justin Holmgren, Alex Lombardi, and Ron D. Rothblum [HLR21].

Our results touch on independently important areas such as non-interactive zero knowledge, delegation of computation, the insecurity of parallel repetition (Question 1.6), and the cryptographic hardness of computing Nash Equilibria in game theory. The broader community has also built on the ideas in this thesis, leading to vibrant lines of work that use provable Fiat-Shamir instantiations to build exciting new cryptographic protocols (e.g. [PS19, BFJ+20, GJJM20, BKM20, LNPY20, JJ21, JKKZ21, CJJ21a, CJJ21b, HJKS22]).

In the rest of this introduction, we first summarize our results appearing in Parts I to III, we give an overview of (some of) our methodology and techniques, and we briefly discuss some conclusions and open problems.

1.3.1 Results in Part I

Part I, titled "The Basic Framework and Initial Constructions," builds families of correlation-intractable hash functions and instantiates the Fiat-Shamir heuristic (for certain protocols of interest) making use of *simple*, *clean* computational assumptions following the "win-win" framework of [GM84]. In Chapter 2 we make use of stronger-than-standard assumptions that we call "(Quantitatively) Optimal Security Assumptions," while in Chapter 3 we rely on truly standard assumptions. Some noteworthy implications are as follows:

- We build succinct non-interactive arguments for (logspace-uniform) bounded depth computation from optimal hardness assumptions related to the learning with errors (LWE) problem. This is the first publicly-verifiable succinct non-interactive argument that does not rely on knowledge assumptions or otherwise unfalsifiable assumptions.[2]

- We build a non-interactive zero-knowledge proof system for NP based on the circular-security of the LWE assumption (as used to build fully homomorphic encryption [BV11]). This is the first lattice-based NIZK for NP, despite the otherwise unreasonable effectiveness of lattice-based cryptogrpahy [Pei16] and the fact that other approaches to achieving NIZKs (without Fiat-Shamir) were known in other contexts [BFM88, FLS90, CHK03]. Finally, this also constituted the first provable instantiation of the Fiat-Shamir heuristic based on a standard cryptographic assumption.

  Our result was improved by a follow-up work of Peikert and Shiehian [PS19] to rely on the *plain* LWE assumption (without circular security).

[2] Concurrent with another approach developed in [KPY18, KPY19].

This part contains the results of [CCH$^+$18] and [CLW18], which appeared together in [CCH$^+$19].

1.3.2 Results in Part II

In Part II, titled "CI Self-Reductions and Further Applications to Protocols," we build additional non-interactive protocols using variants of the Fiat-Shamir heuristic based on LWE. Unlike Part I, which builds CI hash families "from scratch," this part mainly proceeds by using the technical results of Part I (about correlation intractability) as a black box to achieve further results about cryptographic proofs. Relatedly, we expand the class of *relations* with provable instantiations of correlation intractability via forms of *self-reduction* (building more complex CI from simple CI).

This part contains the results of [LV20a, HLR21, LVW19], and includes the following implications:

- We prove that the complexity class PPAD [Pap94] is hard-on-average, assuming on the sub-exponential hardness of LWE along with the hardness of iterated squaring modulo a composite [RSW96]. This implies the cryptographic hardness of computing a Nash equilibrium in bimatrix games [DGP06, CDT09], resolving a long-standing open problem in complexity theory. This was the first construction of PPAD-hardness based on standard cryptographic assumptions.[3] We additionally build a *verifiable delay function* (VDF) [BBBF18] additionally assuming the *sequential* hardness of iterated squaring, which constitutes the first VDF whose security is based on standard (up to the necessary assumption of sequential hardness) cryptographic assumptions.

- We substantially generalize the results of [CCH$^+$19] to apply to a much broader class of interactive proof systems. One major downstream implication is a fairly comprehensive answer to a surprisingly basic open problem about the *original* zero-knowledge proof systems of [GMR85, GMW86, Blu86]:

[3] Concurrent with [KPY20], which relies on a new but reasonable assumption about bilinear maps.

Question 1.6. *Do these proof systems remain zero knowledge when executed* ***many times in parallel?***

In particular, we show that essentially any "commit-and-open" protocol[4] (including the [GMW86] 3-coloring protocol) *fails* to remain zero knowledge under parallel repetition.

These techniques have already been used to great effect: in a recent work [CJJ21b], Choudhuri, Jain, and Jin showed how to build succinct non-interactive arguments for *all polynomial-time computation* by relying on a Fiat-Shamir instantiation leveraging these new ideas.

- We build (based on the LWE assumption) 2-message witness indistinguishable (WI) arguments for NP that are *publicly verifiable*; the argument system consists of a single verifier message followed by a single prover message, and anyone can verify a proof given only the transcript. These are quite related to NIZK proofs/arguments, but also require a form of security against *malicious* verifiers.

1.3.3 Results in Part III

Part III, titled "Multi-Input Correlation Intractability," studies (in large part) forms of CI that reason about *multiple* hash function input-output pairs simultaneously. However, this part also contains key insights regarding the single-input case and the Fiat-Shamir heuristic.

As discussed earlier, one of the most basic properties one might desire from a hash function is *collision resistance.* As such, the following problem has received much attention in theoretical cryptography.

Question 1.7. *What are the assumptions from which collision-resistant hash functions can be built? In particular, can they be built from an arbitrary one-way function?*

Collision resistance can also be viewed as a simple special case of multi-input CI (Definition 1.4). However, more complex forms of multi-input CI are *far* less

[4] This captures a wide class of basic 3-message protocols; our result holds for specific (natural) choices of "commitment scheme" for these protocols.

understood; indeed, most of them had no instantiations based on standard (or even reasonable) assumptions at all!

This part contains the results of [HL18, LV20b], including the following implications:

- We construct various forms of *multi-input* correlation intractable hash functions, including basic primitives such as collision-resistant hash functions, from a quantitatively strong (close to optimally-secure) *one-wayness assumption*. This circumvents a decades-old barrier due to Simon [Sim98]. The construction also generalizes to build hash functions that are correlation-intractable for what we call "efficiently locally samplable relations."

- Specialized to the *single-input* case, we show how to instantiate the Fiat-Shamir heuristic in order to obtain NIZK arguments for NP (based on strong but reasonable assumptions). This was the first usage of "efficient correlation intractability" for Fiat-Shamir and was a key idea towards [CCH$^+$18, CLW18] (Part I).

- We develop a new framework for constructing CI hash functions using a cryptographic primitive called *shift-hiding shiftable functions* (SHSFs) [PS18]. This implies a conceptually simple construction of CI for functions based on LWE (as an alternative to [PS19]). Our construction transparently generalizes to achieving new variants of *multi-input* CI based on standard assumptions.

1.4 Techniques

Having stated our main results, we now proceed to give a high-level overview of our approach, starting with a brief discussion of what was previously known.

1.4.1 Correlation Intractability and Fiat-Shamir for Proofs

Despite all of the negative results and barriers towards instantiating the Fiat-Shamir heuristic, there was a known *plausible* (but unrealized and relatively unstudied) point of attack to at least partially resolving Question 1.2 (and therefore Question 1.1). In [DNRS99], it was noted that if a hash family $\mathcal{H}$ is correlation-intractable (Definition 1.4) for *all* sparse single-input ($t = 1$) relations,[5] then it suffices to instantiate the Fiat-Shamir heuristic for a broad class of statistically-sound interactive proofs. It is not precisely stated in [DNRS99] which protocols they had in mind,[6] but here is a folklore argument for the case of compiling 3-message public-coin interactive proofs:

Claim 1.7.1. *If $\mathcal{H}$ is CI for all sparse relations, then it instantiates the Fiat-Shamir heuristic for all 3-message public-coin statistically sound interactive proofs Π.*

Proof sketch. Let transcripts of Π be denoted by (α, β, γ) where x is the statement, α is the first message, β is the (uniformly random) second message), and γ is the third message. For every $x \notin L$, consider the relation $R = R_x$ defined as

$$R = \Big\{(\alpha, \beta) : \exists \gamma : V(x, \alpha, \beta, \gamma) = 1\Big\}.$$

The proof then amounts to two observations:

- The soundness of Π implies that R is sparse (for all false x). This is because if R were not sparse, a computationally unbounded prover would break the soundness of Π by finding an α (given some false statement x) such that $(\alpha, \beta) \in R$ with non-negligible probability over the choice of β. Provided that this event $(\alpha, \beta) \in R$ occurs over the verifier's randomness β, the prover can then find some string γ that tricks V into accepting.

- The R-correlation intractability of $\mathcal{H}$ implies that $\Pi_{\mathrm{FS},\mathcal{H}}$ is computationally sound. This is simply because if an efficient algorithm can convince the $\Pi_{\mathrm{FS},\mathcal{H}}$ verifier to accept a transcript $(x, \alpha, \beta, \gamma)$ for some fixed false statement x, then by definition of $\Pi_{\mathrm{FS},\mathcal{H}}$ it must be the case that $\beta = h(\alpha)$ and $(\alpha, \beta) \in R_x$. □

[5] A single-input relation R is sparse if for all inputs $x \in \{0,1\}^n$, the fraction of outputs y such that $(x, y) \in R$ is a *negligible* function $n^{-\omega(1)}$.

[6] [DNRS99] reported an unpublished observation of Chaum and Impagliazzo and did not provide details.

Claim 1.7.1 was known since the 1990s, but it did not seem to lead to any positive results on the Fiat-Shamir heuristic. Instead, it reduced one tricky problem (instantiating Fiat-Shamir) to another (building CI for all sparse relations). Indeed, some of the barriers referenced above [BDSG$^+$13] can be (and were) formulated as barriers for constructing correlation-intractable hash functions. Looking ahead, it turns out that the hypothesis of Claim 1.7.1 is simply *too strong* for us to be able to derive any useful results out of it.

1.4.2 Our Methodology in a Nutshell

We propose and develop a two-step methodology for obtaining secure Fiat-Shamir instantiations:

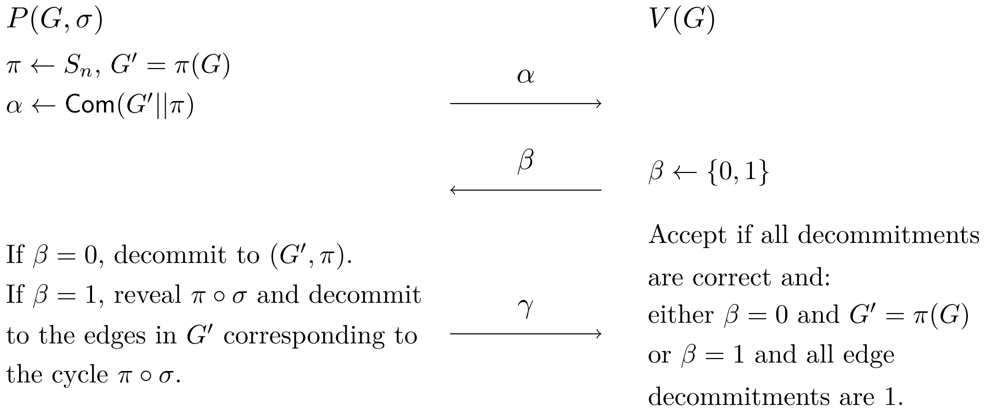
1. For various protocols Π (or protocol classes) of interest, reduce the soundness of (the *specific* protocol) $\Pi_{\mathrm{FS},\mathcal{H}}$ to various *weak* forms of correlation intractability.

2. Build hash functions satisfying these weak forms of correlation intractability from “nice” cryptographic assumptions. We would prefer these assumptions to be falsifiable [Nao03, GW11] or “standard” (e.g., the hardness of learning with errors (LWE, [Reg05])), but will sometimes settle for stronger-than-standard assumptions if they are clean and simple-to-state.

The key insight in formulating this two-step methodology is to restrict the *relation class* subject to our correlation intractability requirement; specifically, we will always consider classes of relations satisfying some *efficiency* properties. This avoids known impossibility results for building correlation intractability and thus makes Step (2) plausible. However, the folklore FS-to-CI reduction (Claim 1.7.1) does not produce relations R that are efficient in any useful sense, so it is a priori unclear how to carry out *either* Step (1) or Step (2). In the end, our results require a careful coordination of what we can construct in Step (2) with what is actually useful in Step (1).

1.4.3 An Example: Fiat-Shamir for the Blum Protocol

To illustrate our approach, we sketch how to instantiate Steps (1) and (2) to obtain a provably secure Fiat-Shamir instantiation for an interactive zero-knowledge proof system for graph Hamiltonicity (due to Blum [Blu86]). We first recall the [Blu86] protocol, where the prover and verifier have an n-vertex graph G and the prover additionally has a permutation $\sigma : [n] \to [n]$ mapping the standard n-cycle to a subgraph of G. The prover and verifier make use of a *commitment scheme* (allowing the prover to commit to bits that are hidden from the verifier until later *opened* by the prover) and execute the following protocol:



$P(G, \sigma)$

$V(G)$

$\pi \leftarrow S_n$, $G' = \pi(G)$

$\alpha \leftarrow \mathsf{Com}(G' || \pi)$

$\xrightarrow{\alpha}$

$\xleftarrow{\beta}$

$\beta \leftarrow \{0, 1\}$

If $\beta = 0$, decommit to (G', π).

If $\beta = 1$, reveal $\pi \circ \sigma$ and decommit to the edges in G' corresponding to the cycle $\pi \circ \sigma$.

$\xrightarrow{\gamma}$

Accept if all decommitments are correct and: either $\beta = 0$ and $G' = \pi(G)$ or $\beta = 1$ and all edge decommitments are 1.

Figure 1-1: The Blum Hamiltonicity Protocol Π_{Blum}

Π_{Blum} as stated only has soundness error 1/2, so we consider the problem of applying Fiat-Shamir to the parallel repeated protocol $\Pi_{\text{Blum}}^{(t)}$ (for some large enough $t \geq \lambda$). As discussed above, Claim 1.7.1 states that if a hash function family $\mathcal{H}$ is CI for all sparse relations, then it can be used to instantiate Fiat-Shamir for Π_{Blum}, but this does not seem useful enough on its own. Instead, let us examine the particular relations R_G that appear in the analysis

$$R_G = \Big\{(\alpha_1, \ldots, \alpha_t, \beta_1, \ldots, \beta_t) : \exists \gamma_1, \ldots, \gamma_t : V(G, \alpha_i, \beta_i, \gamma_i) = 1 \text{ for all } i\Big\}.$$

Reading off from the description of Π_{Blum}, we can interpret that a challenge vector $(\beta_1, \ldots, \beta_t)$ is "bad" for commitments $(\alpha_1, \ldots, \alpha_t)$ (in the sense of being in R_G) if for all i, α_i is a commitment to some $(\pi(G), \pi)$ whenever $\beta_i = 0$, and α_i includes a commitment to a graph containing a cycle whenever $\beta_i = 1$. This is an exact characterization of when the prover can produce accepting $\gamma_1, \dots, \gamma_t$ in the last round.

CI for functions suffices. One important observation is that the relations R_G are *not* as complex as general sparse relations R, in the following sense: every R_G has the property that for all $\alpha_1, \dots, \alpha_t$, there is *at most one* $\beta_1, \dots, \beta_t$ such that $(\alpha_1, \dots, \alpha_t, \beta_1, \dots, \beta_t) \in R_G$. This is essentially a reformulation of what is called the *special soundness* property of Π_{Blum}. What this means is that to obtain a sound Fiat-Shamir instantiation for Π_{Blum}, it would suffice to have a $\mathcal{H}$ that is CI for all *functions*, rather than all sparse relations.

But we still seem stuck. Unfortunately, it still seems difficult to build a correlation-intractable hash family for relations of the form R_G; one serious concern is that the relation R_G (or equivalently, the underlying (partial) function f_G) *cannot be decided in polynomial time.* This seems quite inherent, by the following reasoning:

- Deciding R_G requires understanding what messages are contained in the commitments $\alpha_1, \dots, \alpha_t$, and
- The commitment scheme is supposed to hide the underlying messages from *all polynomial-time algorithms.* This is necessary for the protocol to be zero knowledge.

Solution: Use a Trapdoor! Fortunately (and perhaps surprisingly), there is a resolution that allows R_G to be efficiently decidable without compromising the security of Π_{Blum}. The key point is this: the above reasoning only implies that it must be infeasible for the *verifier* to decide R_G, while we only need to be able to decide R_G in the *soundness security reduction.* Thus, we can take the following approach.

- Instantiate Com using a public-key encryption scheme, so that $\mathsf{Com}(b; r) = \mathsf{Enc}(\mathsf{pk}, b; r)$ for a public key pk sampled as part of the description of the protocol. As long as the encryption scheme is secure, $\mathsf{Com}(b)$ will hide b from the verifier, as desired.

- On the other hand, *soundness* of the protocol will hold *even* in a world where the corresponding secret key sk to pk is known! In this mental experiment, we can forget the (possibly still hard-to-decide) relation $R_{\mathsf{pk},G}$ and instead consider the following modified relation:

$$R'_{\mathsf{pk},\mathsf{sk},G} = \Big\{(\alpha_1, \ldots, \alpha_t, \beta_1, \ldots, \beta_t) : \text{ for all } i, \text{ if } i = 0 \text{ then}$$

$$\mathsf{Dec}(\mathsf{sk}, \alpha_i) = (\pi(G), G) \text{ and if } i = 1 \text{ then } \mathsf{Dec}(\mathsf{sk}, \alpha_i) \text{ contains a cycle } \Big\}.$$

  One can show that if $\mathcal{H}$ is CI for $R'_{\mathsf{pk},\mathsf{sk},G}$ then $\mathcal{H}$ yields a sound Fiat-Shamir instantiation for Π_{Blum}. But now the game has completely changed: given the secret key sk as a trapdoor, $R'_{\mathsf{pk},\mathsf{sk},G}$ is efficiently decidable! Indeed, since it still represents a (partial) function, the conclusion is that $R'_{\mathsf{pk},\mathsf{sk},G}$ represents an efficiently computable (partial) function.

With this idea, we are left with a highly promising approach: to instantiate Fiat-Shamir for Π_{Blum} (for a particular, natural choice of commitment scheme), it now suffices to build a CI hash family for *efficiently computable functions*, rather than for all sparse relations.

CI for Efficiently Computable Functions. So far, we have shown how to instantiate Step (1) of our framework: reducing Fiat-Shamir for Π_{Blum} to a *weak* form of correlation intractability. What remains is Step (2): *constructing* this form of CI. Again, we remark that only fairly trivial forms of single-input CI were known (from standard assumptions such as LWE) before this thesis.

We construct a hash function family $h(k, x)$ with a public hash key k and input x that satisfies correlation-intractability for all "efficiently computable functions" with some fixed polynomial time bound T, meaning the following. For any function f having circuit size T, if a polynomial time adversary is given a random k, it cannot find an input x such that $h(k, x) = f(x)$.

At a high level, the idea of the construction is the following. Designing a hash function $h_f(k,x)$ that is correlation intractable for a single function f is trivial: simply define $h_f(k,x) = f(x)+1$ (or, just flip the last bit of $f(x)$). We will construct a hash function family so that, for *any* f, a random function from the family will look indistinguishable from a hash function that is specifically designed to be correlation intractable with respect to f.

The actual construction is simple, making use of a fully homomorphic encryption (FHE) scheme [Gen09, BV11]:

$$h(k,x) = \mathsf{Eval}_{\mathsf{pk}}(U_x, \mathsf{ct}), \text{ where } k = (\mathsf{pk}, \mathsf{ct}),\ \mathsf{ct} = \mathsf{FHE.Enc}(\mathsf{pk}, g_0),\ \text{and } U_x(g) = g(x).$$

That is, the hash function interprets the hash-key $k = (\mathsf{pk}, \mathsf{ct})$ as a public key pk of an FHE scheme, along with a ciphertext ct encrypting some fixed "dummy" circuit g_0. The hash function homomorphically computes the map $g \mapsto g(x)$ over the ciphertext ct. The key insight is that the function g (initially set to g_0) is completely hidden by the security of the encryption scheme; therefore, one can prove that this $\mathcal{H}$ is CI for a function f by switching ct to be an encryption of some function $g = g_f$ so that $h(k,x)$ is *never* equal to $f(x)$!

Our proof of security is a simple implementation of this intuition. Assume that an adversary gets k and is able to find x such that $h(k,x) = f(x)$ with non-negligible probability. We first switch the ciphertext ct in the key k to be an FHE encryption of the circuit $g(x) = \mathsf{Dec}_{\mathsf{sk}}(f(x)) \oplus 1$, where sk is the FHE secret key. In other words, g first computes $f(x)$, then interprets it as an FHE ciphertext, decrypts it and outputs the opposite bit. We argue that this change is indistinguishable to the adversary by the security of the FHE; this requires circular security since the circuit g depends on sk. Since the adversary cannot distinguish this change, it still outputs x such that $h(k,x) = f(x)$ with non-negligible probability. So, we have:

$$\begin{aligned} f(x) = h(k,x) &= \mathsf{FHE.Eval}_{\mathsf{pk}}(U_x, \mathsf{ct}) \\ &= \mathsf{FHE.Eval}_{\mathsf{pk}}(U_x, \mathsf{Enc}_{\mathsf{pk}}(\langle \mathsf{Dec}_{\mathsf{sk}}(f(\cdot)) \oplus 1 \rangle), \qquad (1.1) \end{aligned}$$

where $U_x(\langle \mathsf{Dec}_{\mathsf{sk}}(f(\cdot)) \oplus 1\rangle) = \mathsf{Dec}_{\mathsf{sk}}(f(x)) \oplus 1$. However, applying $\mathsf{Dec}_{\mathsf{sk}}(\cdot)$ to both sides of (1.1) we get

$$\mathsf{Dec}_{\mathsf{sk}}(f(x)) = \mathsf{Dec}_{\mathsf{sk}}(\mathsf{FHE.Eval}_{\mathsf{pk}}(U_x, \mathsf{Enc}_{\mathsf{pk}}(\langle \mathsf{Dec}_{\mathsf{sk}}(f(\cdot)) \oplus 1\rangle))) = \mathsf{Dec}_{\mathsf{sk}}(f(x)) \oplus 1,$$

where the last equality follows by correctness of $\mathsf{FHE.Eval}$. In other words, once we switched ct to be an encryption of g, we ensured that there is no x for which $h(k, x) = f(x)$.

This completes our sketch of Step (2); combining Steps (1) and (2), we obtain a provably secure Fiat-Shamir instantiation for Π_{Blum} (and thus, in particular, a non-interactive zero-knowledge protocol for NP) relying on a circular-secure FHE scheme. This construction appears in [CLW18] (Chapter 3 of this thesis), and, following [PS19], can even be modified to rely on plain LWE.

1.4.4 Our Methodology, Revisited

The example from Section 1.4.3 is *one* instantiation of our methodology as described in Section 1.4.2. Throughout this thesis, our main results and techniques typically fall into the following two categories:

- We build various kinds of correlation-intractable hash functions from simple (and often standard) computational assumptions. As in Section 1.4.3, it is crucial that the *scope* of our CI hash families is not to handle all sparse relations.

- We show how to *use* these weak CI hash families to instantiate the Fiat-Shamir heuristic for protocols of interest.

In Section 1.4.3 (and Chapter 3), the key variant of CI that we considered was CI for efficiently computable functions. Many of our other results instead work with more powerful forms of CI (that we nevertheless are able to construct). We list some important examples below:

- In Chapter 2, we consider correlation intractability for *efficiently samplable relations*: namely, there is a polynomial time algorithm that given x, samples a uniformly random y such that $(x, y) \in R$. Importantly, for the application to SNARGs, we require a *compact* hash family that does not grow with the running time of the sampler.

- In Chapter 4, we rely on correlation intractability for functions that can be efficiently *guessed* with inverse-subexponential probability.
- In Chapter 5, we work with correlation intractability for relations that are *far* from being functions, but that satisfy nice combinatorial structure arising from (e.g.) parallel repetition.

We refer the reader to the body of the thesis for more details.

1.5 Conclusion and Open Problems

We believe that the general-purpose tools, techniques and abstractions developed in this thesis will help enable a broader understanding of hash functions, the Fiat-Shamir heuristic, and cryptographic proofs. We conclude with a few open research directions related to the progress made in this thesis.

Succinct Arguments. Our progress on correlation intractability and Fiat-Shamir has already led to many new succinct non-interactive arguments for various languages [CCH$^+$18, LV20a, JKKZ21, CJJ21a, CJJ21b, KVZ21]. In particular, we now know how to construct SNARGs for all languages that have non-signalling PCPs. However, it remains wide open to achieve a holy grail of cryptographic proof systems: short, efficiently verifiable non-interactive arguments for *any* NP language.

Question 1.8. *Can we construct succinct non-interactive arguments for all of* NP*?*

Question 1.8 is subject to some limitations [GW11], although it remains unclear how strong these limitations are. On the other hand, the Fiat-Shamir heuristic *predicts* that such arguments exist [Mic94] via Kilian's interactive protocol [Kil92]. Thus, Question 1.8 is also related to questions about the Fiat-Shamir heuristic applied to *argument* systems.

Question 1.9. *Which kinds of interactive* arguments *have sound Fiat-Shamir instantiations?*

Fiat-Shamir for arguments is subject to strong impossibility results [Bar01,GK03], so the landscape here is quite uncertain. For the sake of concreteness, we can be very explicit and ask about Kilian's protocol specifically:

Question 1.10. *Can we instantiate Fiat-Shamir for the [Kil92] protocol?*

A recent work [BBH$^+$19] studies Question 1.10 directly but obtains inconclusive results.

Better Correlation Intractability Even restricted to the setting of Fiat-Shamir for (statistically sound) *proofs*, where correlation intractability is immediately applicable, a number of intriguing directions remain. One very natural direction is understanding *which* sparse relations R we can build CI hash functions for. A conceptually clean breaking point seems to be relations R that are *efficiently decidable.* We ask:

Question 1.11. *Can we build hash functions that are correlation-intractable for all efficiently decidable relations from standard assumptions?*

Another direction is on *compactness.* In all of our constructions based on standard assumptions, the *runtime* of the hash function family $\mathcal{H}$ grows (at least linearly) with the computational complexity of the relation R. Sometimes (such as in Chapter 2) this is *too expensive* of a runtime for the application; one could instead hope for a *compact* family where the runtime of $\mathcal{H}$ is a *fixed* polynomial in its input length.

Question 1.12. *Can we build* compact *CI from standard assumptions?*

One can also hope to minimize the *computational assumptions* required for correlation intractability and Fiat-Shamir. Some recent works [BKM20, JJ21] have successfully built (weak forms of) correlation intractability without relying on the LWE assumption. However, this area remains rather poorly understood. We ask (rather aggressively):

Question 1.13. *Can we build useful forms of CI from any one-way function? Can we build NIZK arguments from any one-way function?*

Again, a recent work [Mou21] studies Question 1.13 but obtains inconclusive results.

Finally, looking back on the example given in Section 1.4.3, we remark that it is somewhat strange that we had to instantiate the *commitment scheme* in the [Blu86] protocol carefully (in particular, to include a trapdoor) in order to obtain provable Fiat-Shamir instantiations. Intuitively, using a commitment scheme *without* a trapdoor should make the scheme *more* secure (rather than potentially less secure), but we currently do not know how to analyze these variants of (e.g.) Blum's protocol. We ask:

Question 1.14. *Can we prove the soundness of Fiat-Shamir for* Π_{Blum} *for* any *choice of commitment scheme?*

We remark in Appendix A (which is from [CLMQ21] Appendix A) that one can provably combine a hash function family $\mathcal{H}$ that is CI for efficiently computable functions with a *random-oracle based* commitment scheme, which has no trapdoor! Of course, the whole point of this thesis is to use *concrete* cryptographic primitives and not resort to heuristic models. Nevertheless, we leave this short result as a philosophical point that, in contrast to "trapdoored" commitment schemes that have been used so far, Blum's protocol using a "maximally unstructured" commitment scheme also admits standard-model Fiat-Shamir hash functions. Nevertheless, Question 1.14 remains wide open.

Progress on any of the listed open questions (and many others) should shed significant light on this emerging area in the theory of cryptography.

Part I

The Basic Framework and Initial Constructions

Chapter 2

Fiat-Shamir from Simpler Assumptions

2.1 Introduction

The Fiat-Shamir transform [FS87] is an attractive template for designing non-interactive argument schemes:

1. Design a potentially highly interactive proof (or argument) system Π in which the verifier is "public-coin", meaning that its only messages are fresh random coins.

2. Compile Π into a two-message protocol Π_{FS}, as follows.

   - The Π_{FS} verifier first sends a description of a "sufficiently complex" hash function h.
   - The Π_{FS} prover responds with the transcript of an emulated execution of Π (including an input x, as well as all messages exchanged between the prover and verifier), in which each verifier message is set to be the value of h applied to the transcript so far.
   - The Π_{FS} verifier checks that the transcript it received is consistent with h, and that the verifier of Π would have accepted.

The resulting protocol Π_{FS} is indeed non-interactive (h can be chosen ahead of time, say as part of a common reference string), it is publicly verifiable, and it adds little in communication and computation. In practice, the Fiat-Shamir transform has been heuristically used as the basis for many important protocols, including identification and signature schemes, publicly-verifiable succinct non-interactive arguments (pv-SNARGs) and NIZKs, e.g. [FS87, PS96, Mic94, BCS16, WTs+18].

A central question in the foundational study of cryptography regards the security of this transformation:

> *For which protocols and hash families does the Fiat-Shamir transform preserve soundness? Under what assumptions can we prove this?*

Security analysis in the random oracle model (ROM) has provided some justification for this design methodology: If h is modeled as a random oracle, then Π_{FS} is sound as long as Π is computationally sound and either has a constant number of rounds [FS87, PS96, AABN02] or more generally, satisfies a stronger soundness property called *soundness against state restoration attacks* [BCS16].

Still, it has remained largely open whether there exist *concrete* hash families that are "FS-compatible" (i.e. that can guarantee soundness and potentially also zero-knowledge for the transformed protocol). Initial results in this direction were negative. Indeed, Goldwasser and Kalai [GK03] (following Barak [Bar01]) demonstrated a three-round, public-coin argument scheme for which applying the Fiat-Shamir transform with *any* hash family never yields a sound protocol. Furthermore, Bitansky et al. [BDG+13] show that, even when starting with a three-round *proof,* soundness of the Fiat-Shamir transform with a concrete hash family cannot be proved via black box reduction to a standard, game-based assumption.

In contrast, a recent line of work [KRR17,CCRR18,HL18] *circumvents* the [BDG+13] impossibility result by using stronger than standard hardness assumptions to construct FS-compatible hash families. Kalai *et al.* [KRR17] gave the first construction of a hash family that is FS-compatible for arbitrary constant-round (public-coin) interactive proofs, albeit from complex obfuscation assumptions. Canetti *et al.* [CCRR18] then provide alternative constructions of FS-compatible hash families without obfuscation, but using complex KDM-security assumptions on secret-key encryption schemes.

We emphasize that the assumptions made by [KRR17, CCRR18] are highly complex in the following sense: both involve an adversary that is in part *computationally unbounded*. For example, the KDM security of [CCRR18] allows messages to be *arbitrary* functions of the key (which may not be efficiently computable). These assumptions are problematic: they are not complexity assumptions [GK16], and they are not falsifiable [Nao03, GW11] except with exponential time. Holmgren and Lombardi [HL18], building on [KRR17], construct a hash family with a different set of serious drawbacks; it relies on indistinguishability obfuscation and is applicable only to a comparatively limited class of protocols.

2.1.1 Our Contributions

We construct explicit hash functions that are FS-compatible for a rich class of protocols, and we prove their security under assumptions that are qualitatively weaker than what was previously known. Using these hash families, we derive new results for delegation of computation and zero knowledge.

We first describe our delegation protocol, which we obtain by applying Fiat-Shamir to the interactive proof of [GKR08] using our new FS-compatible hash functions (and overcoming some technical obstacles that will be further discussed below).

Theorem 2.1 (Informally Stated, see Theorem 2.45)**.** *If any one of the* LWE*-based fully homomorphic encryption schemes in the literature (such as [BV11, BGV12, Bra12, GSW13, BV14]) has optimal security against polynomial-size key-recovery attacks, then there is a publicly verifiable succinct non-interactive argument (*$\mathsf{pv\text{-}SNARG}$*) for (log-space uniform)* NC*. Moreover, there is an efficiently computable hash function family* $\mathcal{H}$ *such that applying the Fiat-Shamir transform to the [GKR08] doubly efficient interactive proof, using* $\mathcal{H}$*, results in such a protocol.*

Here and below, by optimal security against poly-size attacks, we mean that every poly-size circuit family breaks the assumption with probability at most $\lambda^{O(1)}/2^{\lambda}$. We identify a range of the LWE parameters in which this assumption seem plausible. (This range, in particular, involves very high noise magnitude. See further discussion in Section 2.8).

Note that this is the first time that the Fiat-Shamir transform, with an explicit hash function family, is meaningfully applied to an interactive protocol with a super-constant number of rounds. In particular the results of [KRR17, CCRR18, HL18] only hold when the Fiat-Shamir transform is applied to constant-round protocols. See further discussion in Sections 2.1.1 and 2.2.1.

Second, by applying the Fiat-Shamir transform to a specific instantiation of the classical [GMW86] zero-knowledge proof-system we obtain a non-interactive (statistical) zero-knowledge argument for **NP** from a strong variant of LWE:

Theorem 2.2 (Informally Stated, see Theorem 2.53)**.** *If Search-*LWE *is optimally hard for polynomial-size adversaries, then there is a non-interactive zero-knowledge (NIZK) argument system for* **NP** *satisfying either (1) adaptive soundness or (2) statistical zero knowledge. Moreover, there is an efficiently computable hash family $\mathcal{H}$ such that applying the Fiat-Shamir transform to the [GMW86] honest-verifier zero-knowledge proof, using $\mathcal{H}$ (and a specific commitment scheme), results in such a protocol.*

Note that the assumption made in Theorem 2.2 is weaker than that made in Theorem 2.1 as it is directly related to the $\mathsf{Search}\text{-}\mathsf{LWE}$ problem (rather than relying on security of the fully homomorphic encryption schemes which rely on LWE together with a certain circular security assumption). Both assumptions are significantly simpler than those in previous work [KRR17, CCRR18]. In particular, our assumptions do not involve a universal quantifier over computationally unbounded functions.

Note on Adaptively Sound NISZK. We emphasize that our NISZK arguments are only shown to be non-adaptively sound. As is often the case when considering adaptive soundness, the difficulty of using our techniques to get adaptively sound

NISZK arguments stems from the fact that the condition of breaking adaptive soundness is not (in general) efficiently verifiable. Indeed, the work [Pas13] shows that there is no non-black-box reduction from the adaptive soundness of an NISZK argument system to a falsifiable assumption; while we use stronger-than-standard assumptions, this impossibility result still applies to our technique because our NISZK is proven secure via a black-box reduction to an *intermediate assumption* (the assumption that our candidate hash functions satisfy certain kinds of correlation intractability, to be described below) that is falsifiable.

The proofs of both Theorems 2.1 and 2.2 rely on new *correlation intractable hash functions* that we construct as well as new insights on interactive proofs. We next describe these in more detail, since we believe they may be of independent interest. To do so, we first recall the notion of correlation intractability and its relation to Fiat-Shamir.

Correlation Intractability. Loosely speaking, a hash function family $\mathcal{H}$ is correlation intractable (CI) for a sparse relation R if any polynomial size adversary, given a description of $h \leftarrow \mathcal{H}$, outputs x such that $\big(x, h(x)\big) \in R$ with only negligible probability [CGH98]. (A relation is sparse if for every x, the fraction of y's such that $(x, y) \in R$, is negligible.) The hash function is *fully* correlation intractable if it is R-correlation intractable for *all* sparse relations R. Halevi *et al.* [HMR08] observed that if a hash family $\mathcal{H}$ is fully correlation intractable then it is also FS-compatible for every *constant-round* public-coin interactive proof.

Obtaining fully correlation intractable hash functions appears to be quite difficult; as discussed earlier, the only known constructions of such a hash family [KRR17, CCRR18] require assumptions that are not falsifiable except with exponential time. We circumvent this difficulty by focusing on hash families that are correlation intractable for a rich *subclass* of relations. Namely, we consider the class of relations R with the property that it is computationally easy, given an input x, to sample a random output y such that $(x, y) \in R$. We call such relations *efficiently* *sampleable.*

A priori, it is unclear (1) that such hash families are *useful* for obtaining the desired applications, and (2) that they are any easier to construct than fully correlation intractable hash families. The main focus of this work is showing that both of these are actually the case:

- We give new constructions of hash families that are correlation intractable for efficiently sampleable relations, extending the work of [CCRR18]. Crucially, we are able to prove security relying on simple, polynomial time game-based assumptions (albeit with exponentially small winning probability).
- We show that if a hash family $\mathcal{H}$ is correlation intractable for efficiently sampleable relations, then it suffices to instantiate the Fiat-Shamir transform in order to obtain both $\mathsf{pv\text{-}SNARG}$s and NIZKs.

We now describe these two contributions in more detail.

Correlation Intractability for Efficient Relations

We construct two types of efficiently computable hash families that are correlation intractable for the class of efficiently sampleable relations. In our first construction, the (polynomial) complexity of the hash family is allowed to depend on the complexity of sampling the relation.

Theorem 2.3 (Informally Stated, see Theorems 2.13 and 2.25)**.** *If* $\mathsf{Search\text{-}LWE}$ *is optimally hard for polynomial-size circuits, then for every polynomial* $S(\lambda)$*, there is a hash family (whose description size grows with* S*) that is* R*-correlation intractable for all relations that are sampleable by size-*S *circuits.*

This theorem suffices for our construction of NIZK arguments (i.e., Theorem 2.2) because the verifier (which must evaluate a hash function), is *allowed* to run in any polynomial time, even potentially *larger* than the time required by the **NP** verification procedure. In contrast, for our delegation application, we do not know how to use such a non-compact hash function. Rather, we construct a compact correlation intractable hash function (under a stronger assumption).

Theorem 2.4 (Informally Stated, see Theorems 2.13 and 2.20)**.** *If any one of the* LWE*-based fully homomorphic encryption schemes in the literature (such as [BV11, BGV12, Bra12, GSW13, BV14]) has optimal circular security*[1] *against polynomial-size key-recovery attacks, then there exists a hash family that is R-correlation intractable for all relations R that are sampleable by polynomial-size circuits.*

Round-by-Round Soundness

Toward proving Theorem 2.1, we would like to apply the Fiat-Shamir transform to the [GKR08] protocol using the hash function that we constructed in Theorem 2.4. However, we run into a difficulty: correlation intractability is only known to suffice for the Fiat Shamir transform of *constant-round* interactive proofs, whereas the [GKR08] protocol has a super-constant number of rounds.[2]

We overcome this difficulty by formulating a stronger soundness requirement for public-coin interactive proofs that we call *round-by-round (RBR) soundness*. We show that RBR soundness suffices for applying the Fiat-Shamir transform (using a correlation intractable hash function) even for multi-round interactive proofs.[3] To complete the proof of Theorem 2.1, we show that the [GKR08] protocol satisfies RBR soundness and is moreover compatible with our notion of *bounded* correlation intractable hash functions.

[1] The circular security assumption is actually redundant here because all these schemes include an encryption of the secret key to facilitate the bootstrapping procedure [Gen09] and so their security implies that they are also circular secure.

[2] As a matter of fact, there exist statistically sound interactive proofs with a super constant number of rounds (and negligible soundness), to which the Fiat-Shamir transform cannot be applied securely, even in the random oracle model. Consider for example taking the sequential repetition of any interactive proof with constant soundness. While sequential repetition reduces the soundness at an exponential rate, applying the Fiat-Shamir transform (even in the random oracle model) results in an insecure protocol.

[3] We remark that *soundness against state restoration attacks* (which is weaker than RBR soundness) was shown by Ben Sasson *et al.* [BCS16] to suffice for proving soundness of the Fiat-Shamir transform in the *random oracle model*, even for protocols with a super-constant number of rounds. In contrast, we are interested in using Fiat-Shamir in the *plain model* using explicit hash functions, see further discussion in Section 2.2.1.

As a side note, we also show that *any* public-coin interactive proof Π can be easily transformed into an interactive proof that has RBR soundness. The transformation simply applies parallel repetition. As an immediate corollary, fully correlation intractable hash families can be used to transform *any* public-coin, doubly-efficient interactive proof into a publicly verifiable non-interactive argument.

Our main results are summarized in Fig. 2-1:

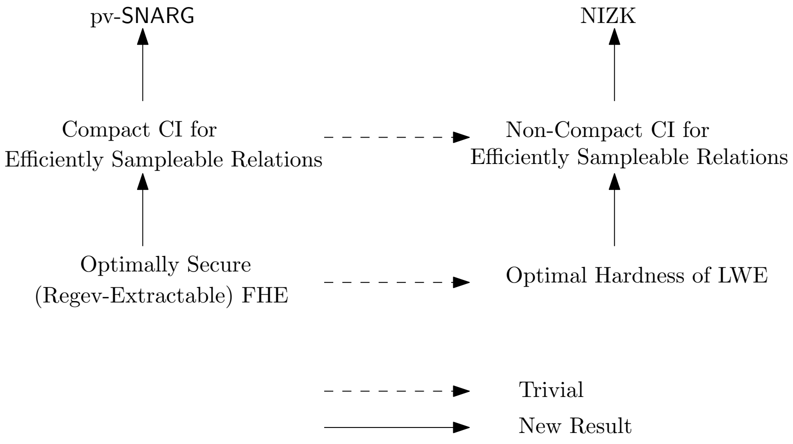

pv-SNARG
NIZK
Compact CI for Efficiently Sampleable Relations
Non-Compact CI for Efficiently Sampleable Relations
Optimally Secure (Regev-Extractable) FHE
Optimal Hardness of LWE
Trivial
New Result

Figure 2-1: Summary of results.

2.1.2 Related Work

On Fiat-Shamir and Magic Functions. Dwork et al. [DNRS99] define *magic functions* to be FS-compatible hash functions for the case of transforming a three-round honest-verifier zero-knowledge argument into a signature scheme, and study the relationship between the existence of magic functions and the existence of general three round zero knowledge protocols.

Correlation Intractability and Fiat-Shamir This work continues a series of recent developments [CCR16, KRR17, CCRR18, HL18] focused on instantiating correlation intractable hash functions in the standard model. We discuss the latter three works, which provide instantiations of FS-compatible hash functions in the standard model.

Kalai *et al.* [KRR17] and Canetti *et al.* [CCRR18] construct correlation intractable hash families from very strong assumptions. Specifically, [KRR17] assumed input-hiding obfuscation for multi-bit point functions and general-purpose indistinguishability obfuscation. Subsequently, [CCRR18] gave a construction that assumed encryption satisfying a form of nearly optimal key-dependent message (KDM) security. More specifically, they assume that polynomial-size adversaries cannot recover the secret key with significantly better probability than random guessing, even given encryptions of *arbitrary* (even inefficiently computable) functions of the secret key. [CCRR18] then give candidate encryption schemes satisfying this security property under strong variants of the LWE and CDH assumptions.

We emphasize that both of these assumptions involve an adversary that is in part *computationally unbounded*. The input-hiding obfuscation in [KRR17] applied to a distribution of point functions

$$P_{\alpha,\beta}(x) = \begin{cases} \beta & \text{if } x = \alpha \\ 0 & \text{otherwise} \end{cases}$$

must hide α even when β is chosen as an arbitrary function of α, and the KDM security of [CCRR18] similarly allows messages to be arbitrary functions of the key. This makes these assumptions difficult to analyze, and in particular they are not falsifiable [Nao03, GW11] except with (non-uniform) exponential time.

A first step towards rectifying this situation was taken by Holmgren and Lombardi [HL18], who consider a *weakening* of full correlation intractability. Their weakening essentially only asks for R-correlation intractability when R is *non-uniformly efficiently sampleable* – there is a circuit of fixed polynomial size that, given x, samples approximately uniformly from the set $\{y : (x,y) \in R\}$. [HL18] constructs this form of "bounded" correlation intractable hash family from a sub-exponentially secure indistinguishability obfuscator and a nearly optimally secure one-way function, and demonstrate that this restricted form of correlation intractability still implies FS-compatibility for the [GMW86] 3-message zero-knowledge proof system for **NP**.

However, their result still requires subexponentially secure indistinguishability obfuscation and has no implications for pv-SNARGs.

pv-SNARGs. Constructions of pv-SNARGs are known in the random oracle model [Mic94], from knowledge assumptions [BCCT13], or from generic assumptions on strong (noiseless) graded encodings with no known candidates [PR17].

A construction of pv-SNARGs was also given by [CCRR18]: they applied the Fiat-Shamir transform (using their hash family) to the [RRR16] constant round interactive proof system for bounded space computation.

In very recent independent work, Kalai *et al.* [KPY18] also construct a publicly verifiable argument system for (logspace uniform) NC. On the positive side, they rely only on falsifiable assumptions about groups equipped with a bilinear map. However, their argument system is in the *preprocessing model*. In this model, the prover and verifier have access to a common reference string, which is as long as the computation transcript (and must be generated securely by a trusted party). In contrast, our protocol requires only a short common random string but relies on seemingly stronger assumptions.

In a later version of their work [KPY19], Kalai *et al.* improve their result to rely on a short common reference string (and extend their pv-SNARGs to work for all polynomial-time computation rather than NC).

Lastly, we remark that *privately-verifiable* (aka designated verifier) non-interactive arguments for all of **P** are known to exist under LWE [KRR14, BHK17].

NIZK Arguments for NP. NIZK arguments for NP are currently known from trapdoor permutations [FLS90], falsifiable assumptions on bilinear maps [GOS06], or indistinguishability obfsucation [SW14, BP15]. The works [GOS06, SW14] also construct NIZK arguments for NP satisfying statistical zero knowledge. Constructing NIZK proofs (or even arguments) for NP from LWE is a long-standing open problem.

Prior works on instantiating the Fiat-Shamir heuristic in the standard model [KRR17, CCRR18, HL18] also give NIZK argument schemes for **NP** under qualitatively stronger assumptions than what is required in this work.

Finally, while not explicitly noted in prior work, combining results of [CCR16, HL18] yields a construction of NIZK arguments (in the common reference string model) from sub-exponentially secure indistinguishability obfuscation and VGB obfuscation. This is the only standard model application of Fiat-Shamir that we are aware of that does not require assuming nearly optimal hardness.

2.2 Our Techniques

We now describe our contributions and high level proof ideas in more detail.

2.2.1 Round-By-Round Soundness

We provide a new soundness definition for interactive proofs that interacts well with the Fiat-Shamir transform. We say that a public-coin interactive proof Π for a language L is round-by-round (RBR) sound if at any stage of the protocol there is a well-defined *state* (depending on the transcript thus far) and some of these states are "doomed"; in the sense that once doomed you will forever remain doomed. More specifically, the first requirement is that for $x \notin L$, the initial state (i.e., corresponding to the empty transcript) is doomed. Second, for every doomed state and every possible next message that a cheating prover might send, with overwhelming probability over the verifier's next message, the protocol state will still be doomed. Lastly, we require that if at the end of the interaction the state is doomed then the verifier will reject (in particular, the state function is efficiently computable on *full* transcripts).

An illustrative example of an interactive proof with round-by-round soundness is the celebrated sumcheck protocol of Lund *et al.* [LFKN90]. Recall that the purpose of the sumcheck protocol is to allow the verifier to check a claim of the form $\sum_{x_1,\ldots,x_m \in \{0,1\}} P(x_1, \ldots, x_m) = \mathbb{v}$, where $P : \mathbb{F}^m \to \mathbb{F}$ is an m-variate polynomial (to which the verifier has oracle access) over a finite field $\mathbb{F}$ and $\mathbb{v} \in \mathbb{F}$ is a fixed field element.[4]

[4] Here and throughout this work we use lowercase blackboard font to denote elements of a finite

The protocol proceeds as follows - the first message from the prover is the (univariate) polynomial $g(\cdot) = \sum_{x_2,\ldots,x_m \in \{0,1\}} P(\cdot, x_2, \ldots, x_m)$. Upon receiving some polynomial $\tilde{g}$ (which may or may not be equal to the prescribed g) from the prover, the verifier checks that it is indeed a low degree polynomial and that $\tilde{g}(0) + \tilde{g}(1) = \mathbb{v}$. Observe that if the initial claim is false, then the prover must send a polynomial $\tilde{g} \not\equiv g$ (or the verifier will immediately reject). Since g and $\tilde{g}$ are low degree polynomials, they must differ on many points. The idea then is for the verifier to select $\mathbb{r}_1 \in \mathbb{F}$ at random and send $\mathbb{r}_1$ to the prover. Since g and $\tilde{g}$ differ on many points, with high probability $\tilde{g}(\mathbb{r}_1) \neq g(\mathbb{r}_1) = \sum_{x_2,\ldots,x_m \in \{0,1\}} P(\mathbb{r}_1, x_2, \ldots, x_m)$. The point is that the latter equation is a sumcheck instance with respect to an $(m-1)$-variate polynomial $P'(x_2, \ldots, x_m) \stackrel{\mathsf{def}}{=} P(\mathbb{r}_1, x_2, \ldots, x_m)$, so the parties recursively run the sumcheck protocol on P'.

To see that the sumcheck protocol has round-by-round soundness we define a partial transcript as doomed if the initial claim for the corresponding step in the recursion is false. As explained above, the sumcheck protocol has the property that at any step of the recursion if we start with a false claim then, with overwhelming probability, we end up with a false claim for the next step in the recursion. This is exactly the meaning of round-by-round soundness. For further details, see Section 2.5.

As one of our contributions, and toward establishing our main delegation result, in Section 2.6 we show that the GKR protocol for log-space uniform NC also has round-by-round soundness.

Round-by-round Soundness and Fiat-Shamir. Our primary motivation for defining round-by-round soundness is to instantiate the Fiat-Shamir transform in the standard model for protocols with a possibly super-constant number of rounds. Indeed, we show that a correlation-intractable hash family suffices for the soundness of the FS transform if the initial protocol is RBR-sound.

To see this, fix any RBR-sound interactive proof Π along with an input $x \notin L$,

field.

and consider the relation:

$$R \overset{\mathsf{def}}{=} \left\{ (\tau, \beta) : \begin{array}{c} \tau \text{ is a doomed partial transcript} \\ \text{and} \\ \tau|\beta \text{ is not doomed} \end{array} \right\}$$

(where τ is a partial transcript ending with a prover message and β is a verifier message).

Round-by-round soundness ensures that R is a sparse relation. Suppose we now apply the Fiat-Shamir transform to the interactive proof, while using a hash function h that is R-correlation intractable. Suppose further that the (computational) soundness of the resulting non-interactive argument is broken. By definition of RBR soundness, this means that the cheating prover has efficiently found some partial transcript τ and verifier message $\beta = h(\tau)$ such that τ is doomed, but (τ, β) is not doomed.[5] Thus, the prover can be used to find a pair $\big(\tau, h(\tau)\big) \in R$, in contradiction to the correlation intractability of the hash function.

Round-by-Round Soundness vs. State Restoration Attacks. A *state restoration attack* [BCS16] on an interactive proof (or more generally an interactive oracle proof) is a cheating prover strategy that is allowed to rewind the protocol to some previous state a limited number of times. Ben Sasson *et al.* showed that soundness against state restoration attacks suffices for compiling interactive proofs using the Fiat-Shamir in the *random oracle model*.[6]

Negligible round-by-round soundness readily implies state restoration soundness for a polynomial number of rewinds. Although it seems reasonable that soundness against state restoration attacks would suffice for instantiating the Fiat-Shamir transform using a correlation interactable hash function (rather than in the random oracle model as shown in [BCS16]), we were unable to prove this.

[5] Such a partial transcript must exist since the empty transcript is doomed, but a full accepting transcript is not doomed.

[6] Prior to the work of [BCS16] this was only shown for *constant-round* interactive proofs [HMR08].

2.2.2 Bounded Correlation Intractable Hash Families

So far, we have shown that correlation intractable hash functions can be used to instantiate the Fiat-Shamir transform for the [GKR08] protocol, yielding pv-SNARGs. In addition, it was already known[7] [CCRR18, HL18] that correlation intractable hash families – with mild additional properties – are also sufficient to yield NIZK argument schemes for NP.

The rest of this work focuses on new constructions of correlation intractable hash families that suffice to yield these applications. These constructions and security reductions all use as a first step (a parameterized version of) the main theorem of [CCRR18] (our Theorem 2.13), which shows how to interpret a secret-key encryption scheme as a correlation intractable hash family if the encryption scheme satisfies two properties (the first being a statistical property and the second a computational one):

1. **Universal Ciphertexts:** An encryption of a random message under any fixed secret key is distributed like an encryption of a random message under a *random* secret key. In particular, this means that ciphertexts are not attached to any one particular key.

2. **Nearly Optimal Bounded-KDM Security against Poly-size Adversaries:** For any function f computable by circuits of a fixed polynomial size, every adversary of arbitrary polynomial size can, given an encryption of $f(k)$ under a (uniformly random) key k, can recover k with probability at most $1/\tilde{\Omega}\left(2^{|k|}\right)$ - i.e., only a polynomial factor better than guessing.

The above property does not suffice to obtain correlation intractable hash families for *all* sparse relations; however, it *does* suffice to obtain hash families that are correlation intractable for all sparse relations that are *sampleable* in some fixed polynomial time. We note that the notion of "bounded correlation intractability" considered in this work is incomparable to that of [CCR16]; they consider correlation intractability for relations that are *decidable* in a fixed polynomial time.

[7] As mentioned earlier, we do improve on previous Fiat-Shamir NIZK instantiations by obtaining statistical zero knowledge, for example.

Since the relations arising from the [GKR08] protocol and a broad class (including [GMW86]) of 3-message zero knowledge proofs for NP satisfy the above notion of efficient sampleability, we have reduced the overall problem to constructing encryption schemes satisfying this weaker notion of bounded-KDM security.

2.2.3 Constructing Optimal Bounded-KDM Secure Encryption

There is a long line of prior work on constructing bounded-KDM secure encryption schemes [BHHO08, ACPS09, BG10, BHHI10, App11]. Unfortunately, the *optimal* level of security stated above that we require is more stringent than was achieved by prior work (which considered any negligible success probability) and poses a significant technical problem, especially when combined with the universal ciphertexts requirement. Still, we show that some of the techniques and instantiations can be adapted to our setting.

Non-Compact CI from Search-LWE. We construct an encryption scheme as above assuming the nearly optimal hardness of search-LWE for poly-time adversaries. Our construction follows the blueprint of [App11], which shows that the class of functions for which an encryption scheme satisfies KDM security can be *amplified* using randomized encodings in the regime of polynomial-size adversaries with inverse polynomial success probabilities.

Recall that a randomized encoding [AIK04] for a function f is a randomized function $\hat{f}$ such that $\hat{f}(x)$ reveals exactly $f(x)$ and nothing else[8] – i.e., there are algorithms $\mathsf{RE.Dec}$ and $\mathsf{RE.Sim}$ such that for all x, $\mathsf{RE.Dec}(\hat{f}(x)) = f(x)$, and $\mathsf{RE.Sim}(f(x)) \approx \hat{f}(x)$. The key point is achieving this so that the function $\hat{f}$ is significantly simpler than f in some way. For example, Yao's garbled circuits [Yao86] are a randomized encoding $\hat{f}$ for any polynomial-time computable f, with the special property that for every r and every input length n, each bit of $\hat{f}(x; r)$ for $x \in \{0,1\}^n$ is a projection of

[8] Technically, $\hat{f}(x)$ may also reveal the input length $|x|$. We will avoid this technicality by, without loss of generality, only considering functions that additionally output the length of their input.

x – that is, either a constant or $x_i \oplus b$ for some fixed $i \in [n]$ and fixed bit b.

Applebaum's idea, following [BHHI10], was to construct an f-KDM secure encryption scheme out of an encryption scheme $\mathcal{E}$ that is $\hat{f}(\cdot\,; r)$-KDM secure for every choice of randomness r. Since $\hat{f}$ is simpler than f, we have made progress. Specifically, the constructed scheme $\mathcal{E}'$ encrypts messages as $\mathcal{E}'.\mathsf{Enc}(m) \overset{\mathsf{def}}{=} \mathcal{E}.\mathsf{Enc}(\mathsf{RE.Sim}(m))$, and correspondingly decrypts ciphertexts as $\mathcal{E}'.\mathsf{Dec}(\mathsf{ct}) \overset{\mathsf{def}}{=} \mathsf{RE.Dec}(\mathcal{E}.\mathsf{Dec}(\mathsf{ct}))$. The point is that an adversary for $\mathcal{E}'$ receives $\mathcal{E}'.\mathsf{Enc}\Big(f(k)\Big) \equiv \mathcal{E}.\mathsf{Enc}\Big(\mathsf{RE.Sim}(f(k))\Big) \approx \mathcal{E}.\mathsf{Enc}\Big(\hat{f}(k; r)\Big)$ for some random r, which still "protects" k by the assumed KDM security of $\mathcal{E}$. This construction can also be modified to obtain a (single) encryption scheme that is simultaneously f-KDM secure for all f in a family $\mathcal{F}$. What is needed in this case is (1) a randomized encoding for a universal function $U_{\mathcal{F}}$, that takes as input a description of $f \in \mathcal{F}$ and an input x and outputs $f(x)$, and (2) an encryption scheme $\mathcal{E}$ that is $\hat{U}_{\mathcal{F}}(f, \cdot\,; r)$-KDM secure for every f and r.

Crucially, we observe that the additional properties we require of $\mathcal{E}'$, namely universal ciphertexts and nearly optimal security, are inherited from $\mathcal{E}$ as long as the randomized encoding scheme RE satisfies two additional properties. First (to ensure universal ciphertexts), RE should be *blind* [BLSV18]: for a uniformly random y, $\mathsf{RE.Sim}(y)$ should also be uniformly random. Additionally, RE should be $1/\tilde{\Omega}(2^{|k|})$-secure.

Bounded KDM Security from Binary-Secret Search-LWE. Our first approach for instantiating the above framework is to use point-and-permute garbled circuits [BMR90] in conjunction with the known circular security of binary-secret Regev encryption. Point-and-permute garbled circuits are perfectly blind [BLSV18], they yield a *universal* randomized encoding $\hat{U}$ for all circuits of some fixed polynomial size, and Regev encryption with an appropriate[9] noise distribution also has (perfectly) universal ciphertexts.

In terms of security, the randomized encoding $\hat{U}$ can also be made sufficiently

[9] Specifically, let the modulus q be even, and take the noise distribution to be uniform on the interval $[-q/4, q/4)$. With a limited number of samples (as is the case in our application), Search-LWE with this setting of parameters reduces to the more typical "narrow discrete Gaussian" noise by a "drowning out the noise" technique.

secure if one-way functions exist that are $2^{-\lambda^{\Omega(1)}}$-hard to invert for $\lambda^{O(1)}$-size adversaries. This assumption is in turn implied by our nearly-optimal Search-LWE assumption. As mentioned previously, for any fixed circuit C and randomness r, each bit of $\hat{U}(C, x; r)$ is a projection of x. Regev encryption with binary secrets is known to be KDM-secure with respect to such projections of the key, under the assumption that binary-secret Search-LWE is hard, and the reduction in fact preserves nearly optimal hardness [ACPS09].

Combining point-and-permute garbled circuits with Regev encryption with binary secrets thus yields, for any polynomial $S = S(n)$, an encryption scheme that has universal ciphertexts and is KDM-secure with respect to any size-S computable functions.

Bounded KDM Security from More General Search-LWE One unsettling aspect of the preceding construction is the reliance on binary-secret LWE, a variant for which algorithms empirically perform better [BG14]. Although we are not aware of attacks on binary-secret LWE that are successful enough to refute a nearly-optimal security conjecture, we still wish to base our constructions on a more general setting of parameters.

We do so by turning to the encryption scheme of [ACPS09], a variant of Regev encryption whose KDM security reduces to Search-LWE with a secret distribution in which each coordinate has higher entropy. Specifically, the secret distribution is uniform over $[-\frac{p}{2}, \frac{p}{2})^n$, and the noise distribution is uniform over $[-\frac{q'}{2}, \frac{q'}{2})^\ell$, for a modulus $q = pq'$ and a prime p. Unfortunately, the KDM security of this scheme is with respect to affine functions over $\mathbb{Z}_p$. In particular, this scheme is *not* known to be secure with respect to bit-by-bit encryptions of its secret key.

To address this difficulty, we construct a new blind randomized encoding from sub-exponentially secure one-way functions for the function U_p that takes as input a *boolean* circuit $C : \mathbb{Z}_p^n \to \{0, 1\}^\ell$ (with elements of $\mathbb{Z}_p^n$ encoded in binary), an input $x \in \mathbb{Z}_p^n$, and outputs $C(x)$. Our construction has the property that for any C and any r, the function $\hat{U}_p(C, \cdot\, ; r)$ is $\mathbb{Z}_p$-affine, which renders our construction suitable for amplifying the KDM security of [ACPS09].

Our construction composes two (blind) randomized encodings.

1. Point-and-permute garbled circuits, which give an encoding $\hat{U}_{\mathsf{bin}}$ of the function U_{bin} that maps $(C, x) \mapsto C(x)$, where $C : \mathbb{Z}_p^n \to \{0,1\}^\ell$ is a boolean circuit, and $x \in \mathbb{Z}_p^n$ is an input. The advantage of this scheme is that it supports arbitrary, e.g. high-depth circuits. On the other hand, $\hat{U}_{\mathsf{bin}}(C, \cdot\,; r)$ is a projection of the *binary representation* of x, instead of a $\mathbb{Z}_p$-affine function of x.

2. An encoding $\hat{U}_{\mathsf{proj}}$ for projections $\pi : \mathbb{Z}_p^n \to \{0,1\}$, where for any π, the function $\hat{U}_{\mathsf{proj}}(\pi, \cdot\,; r)$ is affine over $\mathbb{Z}_p$. Such a randomized encoding follows from a (modified) result of [AIK11] (hereafter AIK), which states that any function f computable by a uniform depth-d *arithmetic* circuit (ensemble) $\{C_n : \mathbb{Z}_p^n \to \mathbb{Z}_p\}_n$ has a perfectly secure, perfectly blind randomized encoding $\hat{f}$ such that $\hat{f}(\cdot\,; r)$ is affine over $\mathbb{Z}_p$ for every r. Specifically, we represent π by a vector $\mathbf{e} \in \{0,1\}^{n \cdot \lceil \log p \rceil}$ (with at most one 1) and a bit b such that $\pi(x) = \langle \mathbf{e}, [\![x]\!] \rangle \oplus b$, where $[\![x]\!]$ denotes the binary representation of x. Then we use the AIK encoding of

$$U_{\mathsf{proj}}\Big((\mathbf{e}, b), x\Big) = \langle \mathbf{e}, [\![x]\!] \rangle \oplus b = \begin{array}{c} b \cdot \Big(\sum_{i=1}^{n \cdot \lceil \log p \rceil} e_i \cdot [\![x]\!]_i \Big) \\ + \\ (1-b) \cdot \Big(1 - \sum_{i=1}^{n \cdot \lceil \log p \rceil} e_i \cdot [\![x]\!]_i \Big). \end{array} \tag{2.1}$$

   U_{proj} is computable by a depth $O(\log n + \log p)$ and size $\tilde{O}(n \cdot p)$ arithmetic circuit over $\mathbb{Z}_p$ by applying the formula

$$[\![x_j]\!]_k = \sum_{y \in \mathbb{Z}_p : [\![y]\!]_k = 1} \Big(1 - (y - x_j)^{p-1} \Big)$$

   to compute each $[\![x]\!]_i$.

A first attempt at composition defines[10] $\hat{U}_p(C, x; r_{\mathsf{proj}}, r_{\mathsf{bin}}) \overset{\mathsf{def}}{=} \hat{U}_{\mathsf{proj}}\Big(\hat{U}_{\mathsf{bin}}(C, \cdot\,; r_{\mathsf{bin}}), x; r_{\mathsf{proj}}\Big)$,

[10] Here, we abuse notation in two ways. First, we write $\hat{U}_{\mathsf{bin}}(C, \cdot\,; r_{\mathsf{bin}})$ to denote the *descriptions* of the corresponding projection functions. Second, we allow $\hat{U}_{\mathsf{proj}}$ to take as input these *multiple*

but this (with the natural simulator) is not blind. The issue is that the simulator for $\hat{U}_{\mathsf{bin}}$ produces a uniformly random string with alphabet $\{0,1\}$, but the AIK simulator for $\hat{U}_{\mathsf{proj}}$ requires a uniformly random string with alphabet $\mathbb{Z}_p$ for its output to be uniformly random (also with alphabet $\mathbb{Z}_p$).

To remedy this, we modify $\hat{U}_{\mathsf{proj}}$. To start, we partition $\mathbb{Z}_p$ into two sets of nearly equal size, $\mathbb{Z}_p = P^{(0)} \sqcup P^{(1)}$, and define a function U'_{proj} that, compared to U_{proj} takes two additional inputs $r^{(0)}$ and $r^{(1)}$. On input $\big((\mathbf{e}, b, r^{(0)}, r^{(1)}), x\big)$, U'_{proj} outputs $r^{(\langle \mathbf{e}, [\![x]\!]\rangle \oplus b)}$ (this can be done by a low-depth circuit analogous to Eq. (2.1)). We then redefine $\hat{U}_{\mathsf{proj}}$ so that $\hat{U}_{\mathsf{proj}}\big((\mathbf{e}, b), x\big)$ samples $r^{(0)} \leftarrow P^{(0)}$ and $r^{(1)} \leftarrow P^{(1)}$, then returns the AIK encoding $\hat{U}'_{\mathsf{proj}}\big((\mathbf{e}, b, r^{(0)}, r^{(1)}), x\big)$. The new decoder for $\hat{U}_{\mathsf{proj}}$ evaluates the AIK decoder for $\hat{U}'_{\mathsf{proj}}$, obtaining $y' \in \mathbb{Z}_p$, and outputs b if $y' \in P^{(b)}$. The new simulator for $\hat{U}_{\mathsf{proj}}$ on input b samples $y' \leftarrow P^{(b)}$, and then returns the output of the AIK simulator for $\hat{U}'_{\mathsf{proj}}$ on y'.

This nearly completes the description of our randomized encoding, except for one subtle issue. For any odd prime p, it is impossible for a partition $\mathbb{Z}_p = P^{(0)} \sqcup P^{(1)}$ to be exactly balanced. This causes the randomized encoding to only be *approximately* blind, where our notion of approximation is the *Renyi divergence* (rather than statistical difference) between the simulator output distribution and the uniform distribution. To suitably decrease the approximation error, we need to replace $\mathbb{Z}_p$ by $\mathbb{Z}_p^k$ for a sufficiently large k.

A Compact Family From FHE. While the above hash families suffice to obtain NIZK argument schemes, they *do not* yield pv-SNARGs when combined with the [GKR08] protocol. This is because in the above hash family, the description of a hash function (and the complexity of hashing) grows polynomially with the complexity of the sampling algorithms of the relations R for which correlation intractability holds.[11] In order to obtain pv-SNARGs, we require a hash family (corresponding

projection functions. We let $\hat{U}_{\mathsf{proj}}((\pi_1, \ldots, \pi_m), x)$ denote the product distribution $\hat{U}_{\mathsf{proj}}(\pi_1, x) \times \cdots \times \hat{U}_{\mathsf{proj}}(\pi_m, x)$.

[11] In the case of [GKR08], we can only give a sampling algorithm that runs in time $\mathsf{poly}(T)$, which ruins succinctness.

to an encryption scheme) that is $\mathsf{SIZE}(S)$-correlation intractable, but yet consists of functions that are evaluable in time much less than S. We in fact construct something stronger – a single hash family that is correlation intractable against all relations that are sampleable by polynomial-size circuits.

This construction also adapts KDM-security amplification techniques in the literature; instead of using randomized encodings [App11], we use fully homomorphic encryption to amplify KDM-security. In particular, [BHHI10] observe that any *circularly secure* FHE scheme satisfying a strong form of evaluation correctness[12] is also KDM-secure for arbitrary polynomial functions of the secret key. The basic [BHHI10] idea is that an adversary can homomorphically generate encryptions of $f(k)$ from the encryption of k (for efficiently computable functions f).

The [BHHI10] observation suggests the following plan to obtain the CI hash families that we desire: start with a FHE scheme that has universal ciphertexts and (sufficiently strong) circular security, and invoke an appropriately modified [BHHI10] argument. However, there are two major flaws in this plan.

- No fully homomorphic encryption scheme in the literature has (anything remotely resembling) universal ciphertexts. Indeed, all schemes in the literature utilize (at the very least) some form of a low-noise Regev encryption, which itself is very far from having universal ciphertexts. A low-noise Regev ciphertext (A, b) under secret key s has the property that $s^t A - b$ is close to either 0 or $\frac{q}{2}$, and therefore Regev encryption is not universal.

- It is not clear how to adapt the [BHHI10] security reduction (that relies on a generic FHE scheme) to the setting of (near-)optimal security. This is because [BHHI10] relies on a FHE scheme with the following strong correctness property: the distribution $\mathsf{Eval}(f, \mathsf{Enc}(x))$ is statistically indistinguishable from an encryption of $f(x)$. In the setting of near-optimal security, a naive application of the [BHHI10] argument would require an extreme form of this correctness property that does not hold for existing FHE schemes in the literature.

[12] Namely, that an f-evaluated encryption of m is statistically indistinguishable from an encryption of $f(m)$

As a result of these problems, we deviate from the plan above in order to achieve unbounded polynomial correlation intractability. Instead of directly working with a fully homomorphic encryption scheme *in the construction*, we consider secret-key Regev encryption, with secret keys uniformly distributed over a moderately sized interval $[-B, B)^n \subseteq \mathbb{Z}_q^n$, and noise distribution $[-q/4, q/4)$. This setting of parameters (which by design yields a scheme with universal ciphertexts) was proposed by [CCRR18][13], and should be contrasted with the typical Regev encryption scheme in which the secret is uniform in $\mathbb{Z}_q^n$ but the noise must be smaller to allow for correct decryption.

We prove that this encryption scheme satisfies unbounded polynomial KDM-security using some associated FHE scheme *in the security proof.*

The KDM security of Regev encryption with these parameters follows from two main observations.

1. Many natural fully homomorphic encryption schemes (e.g., [BV11, BGV12, Bra12, GSW13, BV14]) contain a *low-noise* instantiation of Regev encryption "embedded" within them. That is, from any homomorphically evaluated ciphertext that decrypts to m under an FHE key s, one can efficiently extract a small-noise *Regev* encryption of m under s. We call this property Regev-extractability.

2. Any Regev ciphertext with small noise (which may be arbitrary and malicious) can be re-randomized to obtain a Regev ciphertext whose noise distribution is statistically approximately uniform over $[-q/4, q/4)^m$.

Combining (1) and (2) yields a multiplicatively advantage-preserving reduction from the KDM security of high-noise Regev to the circular security of the (low-noise) Regev-extractable FHE scheme. At a high level, the reduction works as follows: given FHE-circular ciphertexts $\{\mathsf{ct}_i = \mathsf{FHE.Enc}(\mathsf{sk}, \mathsf{sk}_i)\}_i$, one can homomorphically

[13] [CCRR18] leaves the relationship between B and q to be arbitrary except that $B \le q$; we require that B is significantly smaller than q because we reduce from the security of (necessarily low-noise) FHE schemes. Our notion of optimal security for these FHE schemes can only hold when the secret has less entropy than the noise.

compute a (non-random) FHE-ciphertext corresponding to an arbitrary polynomial function $f(\mathsf{sk})$. Then, a secret-key Regev ciphertext of $f(\mathsf{sk})$ can be extracted from this FHE-ciphertext, and the Regev ciphertext can be re-randomized to obtain an approximately uniform Regev encryption of $f(\mathsf{sk})$. Thus, an algorithm that recovers sk from a Regev encryption of $f(\mathsf{sk})$ with better-than-trivial probability can be used to achieve the same key recovery success for the FHE scheme.

We note that a crucial aspect of the analysis is the use of Renyi divergence rather than total variational (aka, statistical) distance in characterizing the re-randomization sampling error.

2.3 Correlation Intractability from KDM-Secure Encryption

This section recalls the definitions of correlation intractable (CI) hash functions and encryption schemes that are secure against key-dependent message (KDM) attacks, as well as the [CCRR18] construction of CI hash functions from strong KDM secure encryption.

Since this work crucially relies on finer-grained notions of indistinguishability and security against resource bounded adversaries, we first adopt the following notation, which is more fine-grained than the standard one. We say that a game $\mathcal{G}$ (that takes as input a security parameter 1^λ) is $(\mathbb{A}, \mathbb{B})$-hard if for all adversaries $\mathcal{A} = \{\mathcal{A}_\lambda\}_{\lambda\in\mathbb{Z}^+} \in \mathbb{A}$, there exists a bound $\epsilon(\cdot) \in \mathbb{B}$ such that for every $\lambda \in \mathbb{Z}^+$, $\mathcal{A}_\lambda$ wins $\mathcal{G}(1^\lambda)$ with probability at most $\epsilon(\lambda)$. When $\mathbb{A}$ is the set of polynomial-size circuit ensembles, we will omit $\mathbb{A}$ and simply say that $\mathcal{G}$ is (computationally) $\mathbb{B}$-hard. If additionally $\mathbb{B}$ is the set of all negligible functions, we simply say that $\mathcal{G}$ is (computationally) hard.

For example, we say that two distribution ensembles $\{X_\lambda\}$ and $\{Y_\lambda\}$ are $\left(\lambda^{O(1)}, \frac{1}{2} + \frac{1}{\tilde{\Omega}(2^\lambda)}\right)$-indistinguishable if for all polynomial-sized circuit ensembles $\mathcal{A} = \{\mathcal{A}_\lambda\}$, there exists some $\epsilon(\cdot)$ with $\epsilon(\lambda) \leq \frac{\mathsf{poly}(\lambda)}{2^\lambda}$ (where the $\mathsf{poly}(\lambda)$ factor may depend on $\mathcal{A}$) such that the advantage of $\mathcal{A}_\lambda$ in distinguishing X_λ from Y_λ is at most $\epsilon(\lambda)$.

2.3.1 Correlation Intractable Hash Functions

Definition 2.1. *A* hash family *is a collection* $\mathcal{H} = \{h_\lambda : \mathcal{I}_\lambda \times X_\lambda \to Y_\lambda\}_{\lambda \in \mathbb{Z}^+}$ *of keyed hash functions such that* $\{\mathcal{I}_\lambda\}$ *is uniformly* $\mathsf{poly}(\lambda)$*-time sampleable and* $\{h_\lambda\}$ *is uniformly* $\mathsf{poly}(\lambda)$*-time evaluable.*

We will also write $\mathcal{H}_\lambda$ *to denote the distribution on functions* $h_\lambda(I, \cdot)$ *obtained by sampling* $I \leftarrow \mathcal{I}_\lambda$*.*

The above definition details the *functionality* of a hash function; there are several security notions that one could require. We focus on (single input) correlation intractability, as put forth by Canetti *et al.* [CGH98].

Definition 2.2 (Correlation Intractability)**.** *For a hash family* $\mathcal{H} = \{h_\lambda : \mathcal{I}_\lambda \times X_\lambda \to Y_\lambda\}_{\lambda \in \mathbb{Z}^+}$ *and a relation ensemble* $R = \{R_\lambda \subseteq X_\lambda \times Y_\lambda\}$*, the* correlation intractability game $\mathcal{G}^{\mathsf{CI}}_{\mathcal{H},R}$ *is the following game, played by any adversary* $\mathcal{A}$ *against a fixed "challenger"* $\mathcal{C}$*:*

1. *On input* 1^λ*,* $\mathcal{C}$ *samples* $I \leftarrow \mathcal{I}_\lambda$ *and sends* I *to* $\mathcal{A}$*.*
2. $\mathcal{A}$ *sends* $x \in X_\lambda$ *to* $\mathcal{C}$*, and wins the game if* $\big(x, h_\lambda(I, x)\big) \in R_\lambda$*.*

We say that $\mathcal{H}$ *is* R-correlation $(\mathbb{A}, \mathbb{B})$-intractable *if* $\mathcal{G}^{\mathsf{CI}}_{\mathcal{H},R}$ *is* $(\mathbb{A}, \mathbb{B})$*-hard.*

Correlation intractability is a useful and versatile property of random oracles that we would like to guarantee in the standard model. However, even a random oracle is only R-correlation intractable for *sparse* relations R.

Definition 2.3 (Sparsity)**.** *For any relation ensemble* $R = \{R_\lambda \subseteq X_\lambda \times Y_\lambda\}$*, we say that* R *is* $\rho(\cdot)$-sparse *if for* $\lambda \in \mathbb{Z}^+$ *and any* $x \in X_\lambda$*,*

$$\Pr_{y \leftarrow Y_\lambda}\Big[(x, y) \in R_\lambda\Big] \leq \rho(\lambda).$$

When ρ *is a negligible function, we say simply that* R *is* sparse*.*

An important complexity measure of a relation R for the purpose of achieving correlation intractability is the complexity of *sampling* from the relation. More formally, we define (following [HL18]) what it means for a relation R to be *efficiently (approximately) samplable.*

Definition 2.4. *A distribution P* multiplicatively ϵ-approximates *a distribution Q if for all outcomes ω, it holds that $P(\omega) \geq \epsilon \cdot Q(\omega)$.*

We note that if P multiplicatively ϵ-approximates a distribution Q, then it also holds for all *events* E, that $P(E) \geq \epsilon \cdot Q(E)$.

Definition 2.5 (Approximate Samplability of Relations)**.** *A relation ensemble $R = \{R_\lambda \subseteq X_\lambda \times Y_\lambda\}$ is* non-uniformly efficiently ϵ-approximately samplable *if there is a* $\mathsf{poly}(\lambda)$*-sized circuit ensemble $\{\mathsf{Samp}_\lambda\}$ such that for every $(x, y) \in R_\lambda$, the distribution $\mathsf{Samp}_\lambda(x)$ multiplicatively ϵ-approximates the uniform distribution on the (by assumption, non-empty) set $\Big\{y' \in Y_\lambda \ : \ (x, y') \in R\Big\}$.*

We say that R is (non-uniformly) efficiently approximately samplable if it is non-uniformly ϵ-approximately samplable for some $\epsilon \geq \frac{1}{\mathsf{poly}(n)}$.

Remark 2.6 (Domain Translation)**.** *Throughout this paper, we make use of the following fact: if $\mathcal{R}$ is a sparse ensemble of relations $\{R_\lambda \subseteq X'_\lambda \times Y_\lambda\}$, then the ensemble $\mathcal{R}'$ obtained by viewing each R_λ as a subset of $X_\lambda \times Y_\lambda$ via some embedding $f_\lambda : X'_\lambda \to X_\lambda$ is also sparse. Moreover, if $\mathcal{R}$ is efficiently sampleable and if $\{f_\lambda^{-1}\}$ is efficiently sampleable, then $\mathcal{R}'$ is also efficiently sampleable.*

This result is used implicitly, e.g. to view a correlation-intractable hash family mapping $\mathbb{Z}_p^n \to \{0,1\}^\ell$ as a correlation-intractable hash family mapping $\{0,1\}^{n \cdot \lfloor \log p \rfloor} \to \{0,1\}^\ell$.

2.3.2 Encryption Schemes and Key-Dependent Message (KDM) Security

Definition 2.7. *A* secret-key encryption scheme (SKE) $\mathcal{E}$ with message space $\mathcal{M} = \{\mathcal{M}_\lambda\}$ *consists of* $\mathsf{poly}(\lambda)$*-time sampleable key distributions $\{\mathcal{K}_\lambda\}_\lambda$ along with* $\mathsf{poly}(\lambda)$*-time computable functions* Enc *and* Dec *(where* Enc *is probabilistic) such that when* *sampling* $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathcal{K}_\lambda$*, it holds with probability* 1 *for all* $m \in \mathcal{M}_\lambda$ *that* $\mathsf{Dec}\Big(\mathsf{sk}, \mathsf{Enc}(\mathsf{sk}, m)\Big) = m$.

In the special case that $\mathcal{M}_\lambda = \{0,1\}$ *for every* λ*, we say that* $\mathcal{E}$ *is a* secret-key *bit*-encryption scheme.

Remark 2.8. *In Definition 2.7, we assumed that keys are of the form* $(\mathsf{pk}, \mathsf{sk})$*, where* pk *denotes key information that we wish to make available to adversaries. For instance, for FHE schemes* pk *will include the evaluation key. Still, it is appropriate to refer to these encryption schemes as secret-key because we do not assume that it is possible to encrypt given only* pk*. In cases where* pk *is always the empty string, we will sometimes just write* $\mathsf{sk} \leftarrow \mathcal{K}_\lambda$ *rather than* $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathcal{K}_\lambda$*.*

Definition 2.9. *A secret-key encryption scheme* $\mathcal{E} = (\{\mathcal{K}_\lambda\}, \mathsf{Enc}, \mathsf{Dec})$ *with message space* $\mathcal{M}_\lambda$ *has* α-universal ciphertexts *if for any key* $(\mathsf{pk}^*, \mathsf{sk}^*) \in \mathcal{K}_\lambda$*, the distribution* $\mathsf{Enc}(\mathsf{sk}^*, \mathcal{U}_{\mathcal{M}_\lambda})$ *multiplicatively* $\alpha(\lambda)$*-approximates the distribution* $\mathsf{Enc}(\mathsf{sk}, \mathcal{U}_{\mathcal{M}_\lambda})$*, where* $\mathcal{U}_{\mathcal{M}_\lambda}$ *denotes the uniform distribution on* $\mathcal{M}_\lambda$ *and* $(\mathsf{pk}, \mathsf{sk})$ *is sampled from* $\mathcal{K}_\lambda$*.*

If $\mathcal{E}$ *has* α*-universal ciphertexts for some* $\alpha(\lambda) \geq \lambda^{-O(1)}$*, then we simply say that* $\mathcal{E}$ *has universal ciphertexts.*

Definition 2.10. *A secret-key bit-encryption scheme* $\mathcal{E} = (\{\mathcal{K}_\lambda\}, \mathsf{Enc}, \mathsf{Dec})$ *is said to be* fully homomorphic *if there is a polynomial-time algorithm* Eval *such that when sampling* $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathcal{K}_\lambda$*, then it holds with probability* 1 *for any circuit* $C : \{0,1\}^n \to \{0,1\}$ *and any* $m_1, \ldots, m_n \in \{0,1\}$*, that*

$$\mathsf{Dec}\Big(\mathsf{sk}, \mathsf{Eval}\big(\mathsf{pk}, C, \mathsf{Enc}(\mathsf{sk}, m_1), \ldots, \mathsf{Enc}(\mathsf{sk}, m_n)\big)\Big) = C(m_1, \ldots, m_n).$$

Definition 2.11. *If* $\mathcal{E}$ *is a secret-key encryption scheme* $\Big(\{\mathcal{K}_\lambda\}, \mathsf{Enc}, \mathsf{Dec}\Big)$ *with message space* $\mathcal{M}_\lambda$*, and if* $f = \{f_\lambda : \mathcal{K}_\lambda \xrightarrow{\$} \mathcal{M}_\lambda^{\ell_\lambda}\}$ *is any (potentially probabilistic) function, the* f-KDM security game $\mathcal{G}^{\mathsf{KDM}}_{\mathcal{E},f}$ *is the following game, played by any adversary* $\mathcal{A}$ *against the following fixed challenger* $\mathcal{C}$*:*

1. *On input* 1^λ*,* $\mathcal{C}$ *samples* $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathcal{K}_\lambda$*, computes* $(M_1, \ldots, M_\ell) \leftarrow f_\lambda(\mathsf{sk})$*, computes encryptions* $\mathsf{ct}_i \leftarrow \mathsf{Enc}(\mathsf{sk}, M_i)$ *for each* $i \in [\ell]$*, and sends* $(\mathsf{pk}, \mathsf{ct}_1, \ldots, \mathsf{ct}_\ell)$

to $\mathcal{A}$.

2. *$\mathcal{A}$ outputs sk', and wins if $\mathsf{sk}' = \mathsf{sk}$.*

$\mathcal{E}$ is said to be f-KDM $(\mathbb{A}, \mathbb{B})$-secure if $\mathcal{G}^{\mathsf{KDM}}_{\mathcal{E},f}$ is $(\mathbb{A}, \mathbb{B})$-hard. If $\mathcal{F}$ is a set of (potentially probabilistic) functions then we say that $\mathcal{E}$ is $\mathcal{F}$-KDM $(\mathbb{A}, \mathbb{B})$-secure *if $\mathcal{E}$ is f-KDM $(\mathbb{A}, \mathbb{B})$-secure for all $f \in \mathcal{F}$.*

2.3.3 Correlation Intractability from Strong KDM Security

In this section, we recall the generic transformation of [CCRR18] and state a stronger version of their main theorem (that follows from their security proof). The differences are explained immediately after the theorem statement.

Construction 2.12 (CCRR Hash Family)**.** *Let $\mathcal{E} = (\{\mathcal{K}_\lambda\}, \mathsf{Enc}, \mathsf{Dec})$ be any secret key encryption scheme with message space $\{0,1\}^\ell$ for $\ell = \ell(\lambda)$. The* CCRR hash family *associated to this encryption scheme, denoted $\mathcal{H}^{\mathcal{E}}_{\mathsf{CCRR}}$, is*

$$\mathcal{H}^{\mathcal{E}}_{\mathsf{CCRR}} = \left\{ h_\lambda : \mathcal{I}_\lambda \times \mathcal{K}_\lambda \to \{0,1\}^\ell \right\}_\lambda$$

where

$$h_\lambda(C, x) := \mathsf{Dec}(x, C),$$

and $\mathcal{I}_\lambda$ is the distribution of ciphertexts C obtained by sampling $K \leftarrow \mathcal{K}_\lambda$, $M \leftarrow \{0,1\}^\ell$, and $C \leftarrow \mathsf{Enc}(K, M)$.

The following theorem, which is based on [CCRR18], shows that the hash family associated with any encryption scheme that (1) has universal ciphertexts (see Definition 2.9) and (2) is exponentially KDM secure (see Definition 2.11), is suitable for the Fiat-Shamir transform.

Theorem 2.13. *Let $\mathcal{E} = (\{\mathcal{K}_\lambda\}, \mathsf{Enc}, \mathsf{Dec})$ be a secret key encryption scheme with $\alpha(\lambda)$-universal ciphertexts, message space $\mathcal{M}_\lambda = \{0,1\}^{\ell(\lambda)}$, and key space $\mathcal{K}_\lambda$ equal to the uniform distribution on $\{0,1\}^{\kappa(\lambda)}$ for some $\ell(\cdot), \kappa(\cdot)$. If $\mathcal{E}$ is $\mathcal{F}$-KDM $(\mathbb{A}, \mathbb{B})$-secure and $R = \{R_\lambda \subseteq \{0,1\}^{\kappa(\lambda)} \times \{0,1\}^{\ell(\lambda)}\}$ is a ρ-sparse relation ensemble that* *is $\beta(\lambda)$-approximately $\mathcal{F}$-sampleable, then $\mathcal{H}^{\mathcal{E}}_{\mathsf{CCRR}}$ is R-correlation $\left(\mathbb{A}, \mathbb{B} \cdot \frac{2^{\kappa(\lambda)} \cdot \rho(\lambda)}{\alpha(\lambda) \cdot \beta(\lambda)}\right)$-intractable.*

Remark 2.14. *There are two main differences between Theorem 2.13 and the original statement in [CCRR18].*

- *Theorem 2.13 parameterizes what KDM functions are required in order to prove correlation intractability for a given relation R in terms of its (approximate) samplability.*

- *Theorem 2.13 assumes a weaker notion of "universal ciphertexts" (Definition 2.9) as compared to [CCRR18].*

However, Theorem 2.13 follows directly from the proof given in [CCRR18], and our proof is included only for completeness.

Proof of Theorem 2.13. Let $\mathcal{E}$ and R be as in the hypothesis of the theorem. Let $\mathcal{H}^{\mathcal{E}}_{\mathsf{CCRR}} = \left\{h_\lambda : \mathcal{I}_\lambda \times \mathcal{K}_\lambda \to \{0,1\}^{\ell(\lambda)}\right\}_\lambda$ be as in Construction 2.12.

Suppose that there is an adversary $\mathcal{A} \in \mathbb{A}$ that, given $I \leftarrow \mathcal{I}_\lambda$, finds an input $x \in \mathcal{K}_\lambda$ such that $\left(x, h_\lambda(I,x)\right) \in R_\lambda$ with probability $\epsilon(\lambda)$ that, for every $\delta \in \mathbb{B}$, satisfies $\epsilon(\lambda) > \delta(\lambda) \cdot \frac{2^{\kappa(\lambda)} \cdot \rho(\lambda)}{\alpha(\lambda) \cdot \beta(\lambda)}$ for some λ. Recall that $\mathcal{I}_\lambda$ is the distribution of an encryption of a uniformly random message under a uniformly random key. That is, we have $I \leftarrow \mathsf{Enc}(K, M)$ where K and M denote random variables whose distributions are uniform over $\{0,1\}^{\kappa(\lambda)}$ and $\{0,1\}^{\ell(\lambda)}$, respectively.

Consider independently sampling a uniformly random key $X^* \leftarrow \mathcal{K}_\lambda$. Then, we have that

$$\Pr_{\substack{K, X^* \leftarrow \mathcal{K}_\lambda \\ M \leftarrow \{0,1\}^{\ell(\lambda)} \\ I \leftarrow \mathsf{Enc}(K,M)}} \left[\mathcal{A}(I) = X^* \wedge \left(X^*, h_\lambda(I, X^*)\right) \in R_\lambda\right] = \frac{\epsilon(\lambda)}{2^{\kappa(\lambda)}},$$

because the above expression can be interpreted as the probability that $\mathcal{A}$ wins the correlation intractability game *and* that $\mathcal{A}(I) = X^*$.

The universal ciphertexts property of $\mathcal{E}$ implies that

$$\Pr_{\substack{X^* \leftarrow \mathcal{K}_\lambda \\ M \leftarrow \{0,1\}^\ell \\ I \leftarrow \mathsf{Enc}(X^*, M)}} \Big[\mathcal{A}(I) = X^* \wedge \ \Big(X^*, h_\lambda(I, X^*)\Big) \in R_\lambda\Big] \geq \frac{\epsilon(\lambda) \cdot \alpha(\lambda)}{2^{\kappa(\lambda)}},$$

because the distribution $\mathsf{Enc}(X^*, M)$ multiplicatively $\alpha(\lambda)$-approximates the distribution $\mathsf{Enc}(K, M)$.

Next, we note that for $I \leftarrow \mathsf{Enc}(X^*, M)$, we have that $h_\lambda(I, X^*) \overset{\mathsf{def}}{=} \mathsf{Dec}(X^*, I) = M$ by the perfect correctness of $\mathcal{E}$. Thus if $\Big(X^*, h_\lambda(I, X^*)\Big) \in R_\lambda$, then $(X^*, M) \in R_\lambda$. Let $S_{x,\lambda}$ denote the set $\{m \ : \ (x, m) \in R_\lambda\}$.

$$\begin{aligned}
\Pr_{\substack{X^* \leftarrow \mathcal{K}_\lambda, \widetilde{M} \leftarrow S_{X^*,\lambda} \\ I \leftarrow \mathsf{Enc}(X^*, \widetilde{M})}} \Big[\mathcal{A}(I) = X^*\Big] &= \sum_x \Pr_{X^* \leftarrow \mathcal{K}_\lambda}[X^* = x] \cdot \Pr_{\substack{\widetilde{M} \leftarrow S_{x,\lambda} \\ I \leftarrow \mathsf{Enc}(x, \widetilde{M})}} [\mathcal{A}(I) = x] \\
&\geq \sum_x \Pr[X^* = x] \cdot \frac{\Pr_{\substack{M \leftarrow \{0,1\}^\ell \\ I \leftarrow \mathsf{Enc}(x, M)}} [\mathcal{A}(I) = x \wedge (x, M) \in R_\lambda]}{\Pr_{M \leftarrow \{0,1\}^\ell}[M \in S_{x,\lambda}]} \\
&\geq \frac{1}{\rho(\lambda)} \cdot \sum_x \Pr[X^* = x] \cdot \Pr_{\substack{M \leftarrow \{0,1\}^\ell \\ I \leftarrow \mathsf{Enc}(x, M)}} [\mathcal{A}(I) = x \wedge (x, M) \in R_\lambda] \\
&= \frac{1}{\rho(\lambda)} \cdot \Pr_{\substack{X^* \leftarrow \mathcal{K}, M \leftarrow U_\lambda \\ I \leftarrow \mathsf{Enc}(X^*, M)}} [\mathcal{A}(I) = X^* \wedge (X^*, M) \in R_\lambda] \\
&\geq \frac{\epsilon(\lambda) \cdot \alpha(\lambda)}{\rho(\lambda) \cdot 2^{\kappa(\lambda)}},
\end{aligned}$$

where $\rho(\lambda)$ denotes the sparsity of $R = R_\lambda$.[14]

Finally, we let $\mathsf{Samp} = \{\mathsf{Samp}_\lambda\} \in \mathcal{F}$ denote an multiplicatively $\beta(\lambda)$-approximate sampler for the relation R, which exists by assumption. We see that

$$\Pr_{\substack{X^* \leftarrow \mathcal{K}, \widetilde{M} \leftarrow \mathsf{Samp}_\lambda(X^*) \\ I \leftarrow \mathsf{Enc}(X^*, \widetilde{M})}} [\mathcal{A}(I) = X^*] \geq \epsilon(\lambda) \cdot \frac{\alpha(\lambda) \cdot \beta(\lambda)}{\rho(\lambda) \cdot 2^{\kappa(\lambda)}},$$

[14] To avoid ambiguity in the case where S_{X^*} is empty, we note that by "$\Pr_{\substack{X^* \leftarrow \mathcal{K}_\lambda, \widetilde{M} \leftarrow S_{X^*,\lambda} \\ I \leftarrow \mathsf{Enc}(X^*, \widetilde{M})}} [f(X^*, \widetilde{M})]$" we actually mean "$\mathop{\mathbf{E}}_{X^* \leftarrow \mathcal{K}_\lambda} \chi(S_{X^*} \text{ is nonempty}) \Pr_{\substack{\widetilde{M} \leftarrow S_{X^*,\lambda} \\ I \leftarrow \mathsf{Enc}(X^*, \widetilde{M})}} [f(X^*, \widetilde{M})]$."

which for all $\delta \in \mathbb{B}$, is greater than $\delta(\lambda)$ for some λ. Thus $\mathcal{A}$ contradicts the assumed $\mathcal{F}$-KDM $(\mathbb{A}, \mathbb{B})$-security of $\mathcal{E}$. Thus, we have proved Theorem 2.13. $\square$

2.4 Optimally KDM-Secure Encryption From Simpler Assumptions

This section presents our two new constructions of KDM-secure encryption schemes from assumptions that are weaker and simpler than previously known. Combined with the results of [CCRR18], recalled in the previous section, this amounts to proving Theorems 2.3 and 2.4 (in Sections 2.4.2 and 2.4.3, respectively).

2.4.1 Learning with Errors

The learning with errors (LWE) problem was introduced by Regev [Reg05]. The following overview is based on Peikert's survey [Pei16].

Definition 2.15 (LWE Distribution). *For any $\mathbf{s} \in \mathbb{Z}_q^n$ and any distribution $\chi \subseteq \mathbb{Z}_q$, the* LWE distribution *$A_{\mathbf{s},\chi} \in \mathbb{Z}_q^n \times \mathbb{Z}_q$ is sampled by choosing $\mathbf{a} \in \mathbb{Z}_q^n$ uniformly at random, sampling $e \leftarrow \chi$, and outputting $(\mathbf{a}, b = \langle \mathbf{s}, \mathbf{a} \rangle + e)$.*

Definition 2.16 (Search LWE). *Let $\ell = \ell(n) \geq 1$, $q = q(n) \geq 2$ be integers, and let $\chi_{\text{sec}}(n)$ and $\chi_{\text{err}}(n)$ be distributions on $\mathbb{Z}_{q(n)}$. The* **Search-LWE**$_{\ell,q,\chi_{\text{sec}},\chi_{\text{err}}}$ *problem, parameterized by n, is to output $\mathbf{s}$ given as input $\ell(n)$ independent samples from $A_{\mathbf{s},\chi_{\text{err}}(n)}$, for $\mathbf{s}$ that is sampled from $\chi_{\text{sec}}(n)^n$.*

For the rest of this paper, we will write LWE in place of **Search-LWE**. All of our lattice based hash functions require (at least) making an assumption of the following form.

Assumption 2.1. *Any $\mathsf{poly}(n)$-time algorithm $\mathcal{A}$ solves* **Search-LWE**$_{\ell,q,\chi_{\text{sec}},\chi_{\text{err}}}$ *with probability at most $\mu(\chi_{\text{sec}})^n \cdot \mathsf{poly}(n, \log(q))$, where $\mu(\chi_{\text{sec}}) := |Supp(\chi_{\text{sec}})|^{-1}$.*

In order for this assumption to have any hope of being true, χ_{sec} must be nearly uniform on its support and it must hold that $\mu(\chi_{\text{err}}) \leq \mu(\chi_{\text{sec}})$ (so that the "error guessing attack" does not violate the assumption). Moreover, when the modulus q is composite, we must take additional care to make sure that "error guessing" modulo factors of q does not break the assumption[15]; see later for more discussion.

In Section 2.8 we describe some basic analysis showing that the best-known polynomial-time algorithms for LWE do not violate our assumption subject to the two conditions above.

Definition 2.17 (Secret-Key Regev Encryption)**.** *For any* $\mathsf{poly}(\lambda)$*-time computable positive integers* $q = q(\lambda) \leq 2^{\mathsf{poly}(\lambda)}$ *and* $n = n(\lambda) \leq \mathsf{poly}(\lambda)$*, and any* $\mathsf{poly}(\lambda)$*-time sampleable distribution ensembles* $\chi_{\mathrm{sec}} = \{\chi_{\mathrm{sec}}(\lambda)\}$ *and* $\chi_{\mathrm{err}} = \{\chi_{\mathrm{err}}(\lambda)\}$ *over* $\mathbb{Z}_{q(\lambda)}$*, we define the encryption scheme* $\mathbf{Regev}_{n,q,\chi_{\mathrm{sec}},\chi_{\mathrm{err}}}$ *to be the secret-key bit-encryption scheme* $\big(\{\mathcal{K}_\lambda\}, \mathsf{Enc}, \mathsf{Dec}\big)$*, where:*

- $\mathcal{K}_\lambda$ *is the distribution* χ_{sec}^n*.*

- *For any* λ *with* $n = n(\lambda)$ *and* $q = q(\lambda)$*, for any* $\mathbf{s} \in \mathbb{Z}_q^n$*, and any* $m \in \{0,1\}$*, the output of* $\mathsf{Enc}(\mathbf{s}, m)$ *is a pair* $(\mathbf{a}, b) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$ *obtained by sampling* $\mathbf{a} \leftarrow \mathbb{Z}_q^n$*, sampling* $e \leftarrow \chi_{\mathrm{err}}(\lambda)$*, and outputting* $(\mathbf{a}, \mathbf{s}^t \cdot \mathbf{a} + m \cdot \left\lceil \frac{q}{2} \right\rfloor + e)$

- $\mathsf{Dec}_\lambda : \mathbb{Z}_q^n \times (\mathbb{Z}_q^n \times \mathbb{Z}_q) \to \{0,1\}$ *is defined so that* $\mathsf{Dec}(\mathbf{s}, (\mathbf{a}, b))$ *is the bit* m *for which* $b - \mathbf{s}^t \cdot \mathbf{a}$ *is closer to* $m \cdot \left\lceil \frac{q}{2} \right\rfloor$ *than to* $(1-m) \cdot \left\lceil \frac{q}{2} \right\rfloor$*.*

A pair $(\mathbf{a}, b) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$ *is a* Regev encryption of $m \in \{0,1\}$ under $\mathbf{s} \in \mathbb{Z}_q^n$ with B-bounded noise *if* $b - \mathbf{s}^t \cdot \mathbf{a} - m \cdot \left\lceil \frac{q}{2} \right\rfloor$ *is in the interval* $[-B, B)$*.*

2.4.2 (P/Poly)-KDM Security via Fully Homomorphic Encryption

In this section, we describe a somewhat generic assumption on the circular security of FHE schemes that implies the existence of a (P/poly)-KDM exponentially-secure encryption scheme with an obliviously sampleable universal ciphertext distribution. Our assumption is *efficiently falsifiable* [Nao03, GW11], albeit with exponentially small

[15] We thank Oded Regev and Noah Stephens-Davidowitz for pointing this out to us.

probability, and is a *complexity assumption* [GK16]. The $(\mathsf{P}/\mathsf{poly})$-KDM secure encryption scheme is simply secret-key Regev encryption (Definition 2.17) where both the secret and the noise distributions are uniform over a relatively large interval in $\mathbb{Z}_q$.

We prove that this scheme achieves $(\mathsf{P}/\mathsf{poly})$-KDM security assuming the security of a LWE-based FHE scheme such as [BV11, BGV12, Bra12, GSW13, BV14] in which *both* the secret and the noise are drawn from the uniform distribution on $[-B, B)$. Our security reduction preserves the kind of exponential security considered in Theorem 2.13, so our assumption can be used as the basis for a candidate correlation intractable hash family.

We now define the notion of homomorphic encryption that suffices for our security reduction. As discussed in Section 2.2, this notion captures FHE schemes whose ciphertexts in some sense "contain" a (low-noise) secret-key Regev ciphertext.

Definition 2.18 (Regev-Extractable Secret-Key Homomorphic Encryption)**.** *A secret-key fully homomorphic bit-encryption* $(\{\mathcal{K}_\lambda\}, \mathsf{Enc}, \mathsf{Dec})$ *with associated homomorphic evaluation algorithm* Eval *is* $\mathbf{Regev}_{n,q,\chi_{\mathrm{sec}}}$-$\mathsf{extractable\ with}$ $B(\lambda)$-$\mathsf{bounded\ noise}$ *(where* $\chi_{\mathrm{sec}}(\lambda)$ *is a distribution on* $\mathbb{Z}_{q(\lambda)}$*) if it satisfies the following structural properties.*

For any $\lambda \in \mathbb{Z}^+$*, denoting* $n = n(\lambda)$*,* $q = q(\lambda)$*, and* $\chi_{\mathrm{sec}} = \chi_{\mathrm{sec}}(\lambda)$*:*

- *The distribution of* $\mathbf{s}$ *when sampling* $(\mathsf{pk}, \mathbf{s}) \leftarrow \mathcal{K}_\lambda$ *is* χ_{sec}^n.
- *There is a polynomial-time algorithm* $\mathsf{Extract}$ *such that:*
  - *For any* λ*, any* $\mathbf{s} \in \chi_{\mathrm{sec}}^n$*, and any* $m \in \{0, 1\}$*, it holds that* $\mathsf{Extract}(\mathsf{Enc}(\mathbf{s}, m))$ *is a Regev encryption* $(\mathbf{a}, b)$ *of* m *under* $\mathbf{s}$ *with* B*-bounded noise, and with* $\mathbf{a}$ *uniformly random in* $\mathbb{Z}_q^n$.
  - *For any* $m_1, \ldots, m_n \in \{0, 1\}$*, any circuit* $C : \{0,1\}^n \to \{0, 1\}$*, and any* $(\mathsf{pk}, \mathbf{s}) \in \mathcal{K}_\lambda$*, it holds with probability* 1 *that*

$$\mathsf{Extract}\Big(\mathsf{Eval}\Big(\mathsf{pk}, C, \mathsf{Enc}(\mathbf{s}, m_1), \ldots, \mathsf{Enc}(\mathbf{s}, m_n)\Big)\Big)$$

is a Regev encryption $(\mathbf{a}, b)$ *of* $C(m_1, \ldots, m_n)$ *under* $\mathbf{s}$ *with* B*-bounded noise.*

We do not assume any particular distribution on the noise of Regev ciphertexts that are extracted from homomorphically evaluated ciphertexts; we assume only that the noise is bounded. For our applications, we require Regev-extractable encryption schemes with the following security property.

Definition 2.19. *Let* $\mathcal{E}$ *be an* FHE *scheme with key distributions* $\{\mathcal{K}_\lambda\}$*. For* $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathcal{K}_\lambda$*, let* $\mathcal{K}_\lambda^{(\mathsf{sk})}$ *denote the distribution of* sk*. Let* $[\![\mathsf{sk}]\!]$ *denote a binary representation of* sk*, and let* $\kappa = \kappa(\lambda)$ *denote the bit-length of such a representation. For any* $\ell = \ell(\lambda)$ *and any class* $\mathbb{B}$ *of functions* $\Delta : \mathbb{Z}^+ \to \mathbb{R}^+$*,* $\mathcal{E}$ *is said to be* [ℓ-bit CPA + circular] $\mathbb{B}$-optimally secure with κ-bit key (abbreviated $(\kappa, \ell, \mathbb{B})$-CCO secure) *if for every* $m_1, \ldots, m_\lambda \in \{0, 1\}$*,*[16] $\mathcal{E}$ *is* f*-KDM* $\left(2^{-\kappa+\mathbb{B}}\right)$*-secure for the "augmented bit-by-bit circular security function"*

$$\begin{aligned} f &= \{f_\lambda : \mathcal{K}_\lambda^{(\mathsf{sk})} \to \{0,1\}^{\ell(\lambda)+\kappa(\lambda)}\} \\ f_\lambda(k) &= m_1 \circ \cdots \circ m_\lambda \circ [\![k]\!] \qquad (\circ \textit{ denotes concatenation}) \end{aligned}$$

Discussion. *The requirement that an encryption scheme is* $(\kappa, \ell, \mathbb{B})$*-CCO secure becomes stronger as* ℓ *or* κ *increases, or as the functions* $\Delta \in \mathbb{B}$ *grow more slowly. In particular, the requirement is trivially satisfied if* $\Delta(\lambda) \geq \kappa(\lambda)$ *for sufficiently large* λ*. This is related to the triviality of constructing a correlation-intractable hash family* $\{\mathcal{H}_\lambda\}$ *in which the output length of* $\mathcal{H}_\lambda$ *is* $O(\log \lambda)$ *– in this case, the only sparse relations are the empty ones.*

Assumption 2.2 (Dream FHE). *For some* n*,* q*,* χ_{sec}*, there* exists[17] *a* $(\kappa, \ell, \mathbb{B})$*-CCO secure secret-key* FHE *scheme that is* $\mathbf{Regev}_{n,q,\chi_{\text{sec}}}$*-extractable with* B*-bounded noise for* $\kappa = \lambda^{\Theta(1)}$*,* $\ell = \lambda^{\Omega(1)}$*,* $\mathbb{B} = \{\Delta \textit{ s.t. } \Delta(\lambda) \leq O(\log \lambda)\}$*,* $B \leq q/\tilde{\Omega}(\lambda)$*, and* χ_{sec}^n *that is sampleable in* $\tilde{O}(n)$ *time using at most* $\kappa + O(\log \lambda)$ *random bits.*

[16] In the case of Regev encryption, and also in our applications with Regev-extractable encryption, we can without loss of generality assume that each message consists entirely of 0's.

[17] In fact, it would even suffice for the construction to be *non-uniform.*

While Assumption 2.2 is not itself falsifiable , the (stronger) assumption that any particular Regev-extractable FHE scheme satisfies Definition 2.19 is a falsifiable (with exponentially small probability) complexity assumption, as claimed.

We also note that the security property postulated in Assumption 2.2 is, even qualitatively, slightly stronger than what is needed for our applications – see the discussion following the proof of Theorem 2.20.

Possible Instantiations of Assumption 2.2 As mentioned earlier, a large family of (secret key variants of) LWE-based FHE schemes – such as [BV11, BGV12, Bra12, GSW13, BV14] are Regev-extractable. Like Regev's encryption scheme, these homomorphic encryption schemes are parameterized by a modulus q, a secret distribution χ_{sec}, and an error distribution χ_{err}. All of these schemes, as written, set χ_{sec} to either be the uniform distribution on $\mathbb{Z}_q$ or a sufficiently wide discrete Gaussian. These distributions are optimal in the polynomial hardness regime [Reg05, ACPS09], but they are trivially *sub*-optimal in the regime of exponential hardness. Specifically, if χ_{sec} is very non-uniform (i.e., a discrete Gaussian), then a key can be directly guessed with probably much better than $2^{-\kappa}$.[18] On the other hand, if χ_{sec} were the uniform distribution over $\mathbb{Z}_q$, then given many Regev ciphertexts (where each ciphertext's noise level is $\frac{q}{4}$-bounded), a secret can be relatively efficiently guessed by first guessing the noise and then computing the secret by linear algebra.

We propose instantiating any of the above-mentioned schemes with secret distribution χ_{sec} and noise distribution χ_{err} such that both are uniformly random on intervals of length ℓ_{sec} and ℓ_{err}, respectively, such that $\ell_{\mathsf{err}} \geq \ell_{\mathsf{sec}}$ and ℓ_{sec} is sufficiently large. We emphasize that, up to a polynomial increase in the modulus-noise ratio, these changes do not affect the polynomial security of the schemes. We are not aware of any algorithm violating Assumption 2.2 for any of these schemes (with the secret distribution as described above), despite the fact that most of the schemes require a superpolynomial (in the case of [BV11], even sub-exponential) modulus-to-noise ratio.

[18] Abstractly, the description length κ is the Shannon entropy of the secret key, while (the negative log of) the trivial guessing probability is the min-entropy. The two entropies agree only for uniform distributions.

However, we note that the scheme [BV14] only relies on a *polynomial* modulus-to-noise ratio in the underlying LWE scheme, which may give us more confidence in the claimed exponential security. We describe the known cryptanalytic results further in Section 2.8.

We are now ready to state our security reduction.

Theorem 2.20. *If Assumption 2.2 is true, then there exist parameters $n = n(\lambda)$, $q = q(\lambda)$, and $\chi_{\mathrm{sec}} = \chi_{\mathrm{sec}}(\lambda)$ such that for some $\ell = \lambda^{\Omega(1)}$, $\mathbf{Regev}_{n,q,\chi_{\mathrm{sec}},\chi_{\mathrm{err}}}$ is $(\mathsf{P}/\mathsf{poly})^{\ell}$-KDM $1/\tilde{\Omega}(2^{\kappa(\lambda)})$-secure where:*

- *$(\mathsf{P}/\mathsf{poly})^{\ell}$ is the class of probabilistic functions that are computable by $\mathsf{poly}(\lambda)$-size probabilistic circuits with ℓ output bits,*
- *χ_{err} is the uniform distribution on $[-q/4, q/4)$, and*
- *κ is the length of the binary representation of an element of χ_{sec}^{n}.*

Our proof of Theorem 2.20 relies on the following lemma, whose proof easily follows from direct computation.

Lemma 2.21. *For any $e \in \mathbb{Z}$ with $|e| \le b$ and for any interval $I = [c, d]$ of length ℓ, the distribution $e + U_{[c-b,d+b]}$ multiplicatively $(\frac{\ell}{\ell+2b})$-approximates the distribution U_I, where for a set S, U_S denotes the uniform distribution on S.*

Proof of Theorem 2.20. Let $\mathcal{E} = (\{\mathcal{K}_\lambda\}, \mathsf{Enc}, \mathsf{Dec}\}$ denote the dream FHE scheme that is (κ, ℓ, Δ)-CCO secure and $\mathbf{Regev}_{n,q,\chi_{\mathrm{sec}}}$-extractable with B-bounded noise for $B \le q/\tilde{\Omega}(\lambda)$. Without loss of generality suppose that $\ell \le \lambda$. Let Eval and $\mathsf{Extract}$ denote the corresponding homomorphic evaluation and extraction algorithms.

Let $\chi_{\mathrm{err}}(\lambda)$ denote the uniform distribution on $[-q/4, q/4)$, and let $(\{\mathcal{K}'_\lambda\}, \mathsf{Enc}', \mathsf{Dec}')$ denote $\mathbf{Regev}_{n,q,\chi_{\mathrm{sec}},\chi_{\mathrm{err}}}$. Suppose for contradiction that $\mathbf{Regev}_{n,q,\chi_{\mathrm{sec}},\chi_{\mathrm{err}}}$ is *not* $(\mathsf{P}/\mathsf{poly})^{\ell}$-KDM δ-secure.

That is, suppose there exist functions $\{f_\lambda : \mathbb{Z}_{q(\lambda)}^{n(\lambda)} \to \{0,1\}^{\ell(\lambda)}\}$ and $\{\mathcal{A}_\lambda\}$, evaluable by $\mathsf{poly}(\lambda)$-size circuits, such that for infinitely many λ,

$$\Pr[\mathcal{A}_\lambda(\mathsf{ct}_1, \ldots, \mathsf{ct}_\ell) = \mathbf{s}] > \delta(\lambda).$$

in the probability space defined by sampling $\mathbf{s} \leftarrow \mathcal{K}'_\lambda$ and, for each $i \in [\ell]$, independently sampling Regev encryptions $\mathsf{ct}_i \leftarrow \mathsf{Enc}'(\mathbf{s}, f_\lambda(\mathbf{s})_i)$.

We will now describe an adversary $\mathcal{B} = \{\mathcal{B}_\lambda\}$, implementable by a $\mathsf{poly}(\lambda)$-sized circuit, that contradicts Assumption 2.2. $\mathcal{B}_\lambda$ is given as input $(\mathsf{pk}, c_1, \ldots, c_{\ell+\kappa})$, and does the following.

1. For $i \in [\ell]$, define $(\mathbf{a}_i, b_i) := \mathsf{Extract}(c_i)$.

2. Compute $(c'_1, \ldots, c'_\ell) := \mathsf{Eval}(\mathsf{pk}, f_\lambda, c_{\ell+1}, \ldots, c_{\ell+\kappa})$ and define $(\mathbf{y}_i, z_i) := \mathsf{Extract}(c'_i)$ for every $i \in [\ell]$.

3. For each $i \in [\ell]$, update $\mathbf{y}_i := \mathbf{y}_i + \mathbf{a}_i$ and $z_i := z_i + b_i$.

4. For each $i \in [\ell]$, sample e_i from the uniform distribution on $[-\frac{q}{4} - 2B, \frac{q}{4} + 2B]$ and update $z_i := z_i + e_i$.

5. Compute and output $\mathcal{A}_\lambda\Big((\mathbf{y}_1, z_1), \ldots, (\mathbf{y}_\ell, z_\ell)\Big)$.

If $(\mathsf{pk}, \mathbf{s})$ is sampled at random from $\mathcal{K}_\lambda$, if $c_{\ell+j} \leftarrow \mathsf{Enc}(\mathbf{s}, [\![\mathbf{s}]\!]_j)$ for each $j \in [\kappa]$, and if $c_i \leftarrow \mathsf{Enc}_\lambda(\mathbf{s}, 0)$, then by the definition of extractability, it holds that after Step 2, each $(\mathbf{y}_i, z_i)$ is a Regev encryption of $f_\lambda(\mathbf{s})_i$ under $\mathbf{s}$ with B-bounded noise, and $(\mathbf{a}_1, \ldots, \mathbf{a}_\ell)$ is uniformly random (and independent of $(\mathbf{y}_1, \ldots, \mathbf{y}_\ell)$). After Step 3, each $(\mathbf{y}_i, z_i)$ is a Regev encryption of $f_\lambda(\mathbf{s})_i$ under $\mathbf{s}$ with $2B$-bounded noise. After Step 4, by Lemma 2.21, it holds that, for

$$\epsilon = \epsilon(\lambda) \overset{\mathsf{def}}{=} \left(\frac{q/2}{q/2 + 2B}\right)^\ell > \left(1 - \frac{4B}{q}\right)^\ell \geq \left(1 - \frac{1}{\tilde{\Omega}(\lambda)}\right)^\lambda = \lambda^{-O(1)},$$

the distribution of $\Big((\mathbf{y}_1, z_1), \ldots, (\mathbf{y}_\ell, z_\ell)\Big)$ multiplicatively ϵ-approximates the distribution on $(\mathsf{ct}_1, \ldots, \mathsf{ct}_\ell)$ obtained by independently sampling $\mathsf{ct}_i \leftarrow \mathsf{Enc}'(\mathbf{s}, f_\lambda(\mathbf{s})_i)$ for each $i \in [\lambda]$. Thus $\mathcal{A}_\lambda$, and therefore $\mathcal{B}_\lambda$, outputs $\mathbf{s}$ with probability at least $\frac{\delta}{\mathsf{poly}(\lambda)} = 2^{-\kappa} \cdot \lambda^{\omega(1)}$. □

Loosely speaking, what our reduction really requires is the ability to re-randomize Regev encryptions in a somewhat weaker sense than what is typically meant by re-randomization. It can receive this ability in the form of Regev ciphertexts. In contrast, (κ, ℓ, Δ)-CCO security gives the reduction even more, specifically fresh $\mathcal{E}$-ciphertexts from which Regev ciphertexts can be extracted. It would instead suffice for $\mathcal{E}$ to satisfy a version of $(\kappa, 0, \Delta)$-CCO security in a setting where the adversary is only given ℓ *Regev* encryptions $\{(\mathbf{a}_i, b_i)\}_{i \in [\ell]}$ for uniform and independent $\{\mathbf{a}_i\}$.

2.4.3 $\mathsf{SIZE}(\kappa^c)$-KDM Security via Randomized Encodings

In this section, we give two additional constructions of encryption schemes satisfying universal ciphertexts (Definition 2.9) as well as $\mathsf{SIZE}(\kappa^c)$-KDM δ-security for $\delta(\lambda) = 2^{-\kappa} \cdot \mathsf{poly}(\kappa)$.[19] These schemes differ from the encryption scheme in Theorem 2.20 in two (related) ways:

- The size bound S for the KDM functions must be specified in advance before choosing the encryption scheme; in contrast, Theorem 2.20 gives a single encryption scheme that (under Assumption 2.2) is KDM-secure for KDM functions that are computable by *any* polynomial-size (probabilistic) circuit.
- Moreover, the encryption schemes in this section are *non-compact*; that is, the size of a ciphertext depends polynomially on the size bound S.

While these schemes satisfy weaker *efficiency* properties than the scheme in Theorem 2.20, we are able to prove *security* based on the exponential hardness of *plain search*-LWE (in contrast to the additional circular security assumptions that were required in Theorem 2.20). Since non-compact (exponential) KDM-secure encryption schemes of the above form suffices to instantiate NIZK arguments in the common random string model (as shown in Section 2.7.2), this yields candidate NIZK arguments based on exponential variants of plain LWE.

To prove our results in this section, we revisit the idea of KDM security amplification via randomized encodings [BHHI10, App11]. In particular, we prove that the

[19] As usual, by this we mean that for every polynomial size adversary $\mathcal{A}$, there exists a constant c such that $\mathcal{A}$ recovers the secret key with probability at most $2^{-\kappa} \cdot \kappa^c$. Recall that $\kappa = \kappa(\lambda)$ denotes the length of a secret key.

generic transformation of [App11] allows us to amplify CCRR-compatibility provided that we use a randomized encoding that is *perfectly blind* (which just means that the simulator applied to a uniformly random string outputs a uniformly random string[20]). By modifying (and composing) standard randomized encoding schemes from the literature [BMR90, IK02, AIK11], we therefore reduce the problem to constructing $\mathcal{F}$-KDM $2^{-\kappa} \cdot \mathsf{poly}(\kappa)$-secure encryption schemes (with universal ciphertexts) for simple function classes $\mathcal{F}$ (namely, some form of affine functions modulo a prime). We then give schemes (based on secret-key Regev encryption or a variant of the [ACPS09] encryption scheme) that satisfy these weaker requirements under an appropriate LWE assumption.

The Generic Transformation

We first recall the generic transformation from [App11] that amplifies (standard) KDM security.

Definition 2.22 (Randomized Encoding)**.** *A randomized encoding scheme for a circuit class $\mathcal{C}$ consists of three algorithms* $(\mathsf{RE.Enc}, \mathsf{RE.Dec}, \mathsf{RE.Sim})$ *with the following syntax.*

- $\mathsf{RE.Enc}$ *takes as input a circuit C and an input x; it outputs an encoding $\langle C, x\rangle$.*
- $\mathsf{RE.Dec}$ *takes as input an encoding $\langle C, x\rangle$; it outputs an evaluation y.*
- $\mathsf{RE.Sim}$ *takes as input a size bound 1^S, a circuit C, and an output y; it outputs an encoding $\tilde{y}$.*

A randomized encoding scheme must satisfy two properties:

- ***Correctness****: for any circuit C and input x, we have that* $\mathsf{RE.Dec}(\mathsf{RE.Enc}(C, x)) = C(x)$ *with probability* 1*.*

[20] We can actually rely on a slightly weaker property defined below.

- ***μ-simulation security**: For every circuit C of size at most S and any input x, the following two distributions are μ-(computationally) indistinguishable.*

$$\mathsf{RE.Enc}(C, x) \approx_{c,\mu} \mathsf{RE.Sim}(1^S, C, C(x)).$$

We say that a randomized encoding scheme is universal *if there is a simulator* $\mathsf{RE.Sim}$ *as above that takes as input only* $(1^S, C(x))$ *and not the circuit* C*.*

Definition 2.23 (Amplified KDM-secure Encryption Scheme). *Let* $\mathcal{E}' = \{(\mathcal{K}'_\lambda, \mathsf{Enc}'_\lambda, \mathsf{Dec}'_\lambda)\}$ *denote a secret key encryption scheme, and let* $\mathsf{RE} = (\mathsf{RE.Enc}, \mathsf{RE.Dec}, \mathsf{RE.Sim})$ *denote a universal randomized encoding scheme for some circuit class* $\mathcal{C}$*. Finally, let* $S = \mathsf{poly}(\kappa)$ *denote some size bound. We then define the* RE*-amplified secret key encryption scheme* $\mathsf{AMP}^{\mathcal{E}'} = \{(\mathcal{K}_\lambda, \mathsf{Enc}_\lambda, \mathsf{Dec}_\lambda)\}$ *as follows.*

- $\mathcal{K}_\lambda$ *is identical to* $\mathcal{K}'_\lambda$*.*
- *The output of* $\mathsf{Enc}_\lambda(\mathsf{sk}, m)$ *is* $\mathsf{Enc}'_\lambda\Big(\mathsf{sk}, \mathsf{RE.Sim}(1^S, m)\Big)$*.*
- *The output of* $\mathsf{Dec}_\lambda(\mathsf{sk}, \mathsf{ct})$ *is* $\mathsf{RE.Dec}\Big(\mathsf{Dec}'_\lambda(\mathsf{sk}, \mathsf{ct})\Big)$*.*

In [App11], it is shown that if $\mathcal{E}'$ satisfies ordinary KDM security with respect to some function class $\mathcal{G}$, and if $\mathcal{F}$ is some function class with circuit representations such that for any $f \in \mathcal{F}$, the function $x \mapsto \mathsf{RE.Enc}(f, x; r)$ lies in $\mathcal{G}$ for any fixed r, then $\mathsf{AMP}^{\mathcal{E}'}$ is KDM secure with respect to $\mathcal{F}$. Our goal is to prove an analogous result that also preserves the conditions of Theorem 2.13, namely nearly optimal security and, more challengingly, the universal ciphertexts property. To do this, we will require randomized encoding schemes satisfying the additional property that we call (a relaxation of) *blindness*, following [BLSV18].

Definition 2.24 (Blind Randomized Encodings). *A randomized encoding scheme* $\mathsf{RE} = (\mathsf{RE.Enc}, \mathsf{RE.Dec}, \mathsf{RE.Sim})$ *is called ϵ-approximately blind for output distribution* χ_{out} *if for any circuit C of size at most S, the following two distributions ϵ-multiplicatively approximate each other:*

1. $\mathsf{RE.Sim}(1^S, C, \chi_{\mathsf{out}})$*.*

2. The uniform distribution on strings of length $\ell' := \left|\mathsf{RE.Sim}(1^S, C, 0^\ell)\right|$*.*

We say that RE *is* perfectly blind *for output distribution* χ *if it is* 1*-approximately blind for* χ*.*

In the context of statistical (or perfect) randomized encodings, [AIK04] refers to such an encoding scheme as *balanced*.

Given this additional property, we are able to state our theorem for this subsection.

Theorem 2.25. *Suppose that* RE *is a universal randomized encoding scheme for a circuit class* $\mathcal{C} \subseteq \mathsf{SIZE}(S)$ *satisfying the following properties.*

- RE *satisfies* $o(2^{-\kappa})$*-simulation security.*
- RE *is* ϵ*-approximately blind for the uniform output distribution, where* ϵ *is any non-negligible function.*
- *For every circuit* $C \in \mathcal{C}$ *and every fixed choice of randomness* r*, the function* $\mathsf{RE.Enc}(C, x; r)$ *is in the class* $\mathcal{G}$*.*

Moreover, suppose that $\mathcal{E}'$ *is an encryption scheme with universal ciphertexts that is* $\mathcal{G}$*-KDM* δ*-secure with uniformly random* κ*-bit keys and message length* ℓ'*. Then, the amplified encryption scheme* AMP *(Definition 2.23) is an encryption scheme for messages of length* ℓ *that has universal ciphertexts and is* $\mathcal{F}$*-KDM secure, where* $\mathcal{F}$ *denotes the class of all functions computable by circuits in* $\mathcal{C}$*.*

Proof. We first prove the universal ciphertexts property; that is, that for any fixed secret key sk, we have that the distribution $\mathsf{AMP.Enc}(\mathsf{sk}, U_\ell)$ multiplicatively $\epsilon = \frac{1}{\mathsf{poly}(\kappa)}$-approximates the distribution $\mathsf{AMP.Enc}(U_n, U_\ell)$. To see this, let Enc'_λ and Dec'_λ denote the encryption and decryption procedures of $\mathcal{E}'$, and note that by the blindness of RE, we have that

$$\mathsf{AMP.Enc}(\mathsf{sk}, U_\ell) \equiv \mathsf{Enc}'_\lambda(\mathsf{sk}, \mathsf{RE.Sim}(1^S, U_\ell)) \succeq_\epsilon \mathsf{Enc}'_\lambda(\mathsf{sk}, U_{\ell'}),$$

where $\succeq_\epsilon$ denotes multiplicative ϵ-approximation. Similarly, we have that

$$\mathsf{AMP.Enc}(U_n, U_\ell) \equiv \mathsf{Enc}'_\lambda(U_n, \mathsf{RE.Sim}(1^S, U_\ell)) \,{}_\epsilon\!\preceq \mathsf{Enc}'_\lambda(U_n, U_{\ell'}).$$

Thus, we conclude that the universal ciphertexts property of AMP follows directly from the same property for $\mathcal{E}'$.

Next, we prove that the transformation also preserves nearly-optimal KDM security. To see this, suppose that for some $f \in \mathcal{F}$, a ppt adversary $\mathcal{A}$ that is given

$$\mathsf{ct} \leftarrow \mathsf{AMP.Enc}(\mathsf{sk}, f(\mathsf{sk})) \equiv \mathsf{Enc}'_\lambda(\mathsf{sk}, \mathsf{RE.Sim}(1^S, f(\mathsf{sk})))$$

returns sk with probability $\delta = \omega(2^{-n})$. Then, by the $o(2^{-n})$-simulation security of RE, the same is true when $\mathcal{A}$ is given

$$\mathsf{ct} \leftarrow \mathsf{Enc}'_\lambda(\mathsf{sk}, \mathsf{RE.Enc}(C, \mathsf{sk}))$$

where $C \in \mathcal{C}$ is some circuit computing f. This will allow us to break the KDM security of $\mathcal{E}'$ for some function $g \in \mathcal{G}$. Namely, an adversary $\mathcal{A}'$ can break the security of $\mathcal{E}'$ by choosing uniformly random encoding randomness r and submitting the KDM function $g(\mathsf{sk}) = \mathsf{RE.Enc}(C, \mathsf{sk}; r)$. By assumption, g lies in the class $\mathcal{G}$, and feeding a SKE-KDM ciphertext ct to $\mathcal{A}$ will result in recovering sk with probability $\delta - o(2^{-\kappa}) = \Omega(\delta)$. This completes the security reduction. □

$\mathsf{SIZE}(\kappa^c)$-KDM Secure Encryption Schemes with Universal Ciphertexts

Together with suitable randomized encoding schemes, Theorem 2.25 reduces the problem of constructing (non-compact) $\mathsf{SIZE}(\kappa^c)$-KDM secure encryption schemes with universal ciphertexts to the problem of constructing $\mathcal{F}$-KDM secure encryption schemes for smaller classes of KDM functions. We follow this recipe with two randomized encoding schemes from the literature, combined with KDM-secure encryption schemes for (two classes of) simple functions. The first construction is straightforward, and assumes the nearly optimal hardness of Search-LWE with binary secrets and a specific noise distribution (uniform on $[-q/4, q/4)$). The second construction is more involved, but allows more general secret and noise distributions.

Point-and-Permute Garbled Circuits. Point-and-Permute garbled circuits, introduced by [BMR90] in order to achieve constant round secure multiparty computation, are a modification of Yao's garbling scheme [Yao86, LP09]; in a nutshell, rather than requiring every entry of a garbled table in Yao's scheme to be decrypted (and that in an honest evaluation only one of the four ciphertexts should be decrypted successfully), point-and-permute garbled circuits augment each wire key $k_{g,b}$ with a random *pointer* $b \oplus r_g$ indicating which table entries $k_{g,b}$ is able to decrypt. While originally introduced in order to allow for a form of distributed garbling [BMR90], and later used for reasons of *efficiency* (i.e. saving a factor of 4 in evaluation time), [BLSV18] noted and took advantage of the fact that point-and-permute garbled circuits are also perfectly blind.

The following theorem follows from the works [BMR90, Rog91, BLSV18]. We refer the reader to [BLSV18] for details on the proof of blindness.

Imported Theorem 2.26. *If one-way functions exist, then there exists a universal randomized encoding scheme* RE *for the class of all polynomial size circuits with the following properties.*

- RE *is perfectly blind.*

- *For any fixed choice of randomness* r *and circuit* C*, the function* $x \mapsto \mathsf{RE.Enc}(C, x; r)$ *is an* $\mathbb{F}_2$*-affine* projection *of* x*. This means that every output bit of* $\mathsf{RE.Enc}(C, x; r)$ *is an* $\mathbb{F}_2$*-affine function of* x *that depends only one bit of* x*.*

- *The function* $(C, x) \mapsto \mathsf{RE.Enc}(C, x; r)$ *is a concatenation* $f_1(C, x; r) || f_2(r)$*, where each bit of* $f_1(C, x; r)$ *has constant input locality.*[21]

Moreover, if subexponentially secure *one-way functions exist, then for any* $c > 0$*,* RE *can be modified so that it is* $2^{-\kappa^c}$*-simulation secure.*

[21] We only use this property in Section 2.4.3.

A Scheme from Exponential LWE with Binary Secrets. Combining Theorem 2.25 with Imported Theorem 2.26, we conclude that to construct a $\mathsf{SIZE}(\kappa^c)$-KDM $2^{-\kappa} \cdot \mathsf{poly}(\kappa)$-secure encryption scheme with universal ciphertexts, it suffices to construct a $\mathcal{F}$-KDM $2^{-\kappa} \cdot \mathsf{poly}(\kappa)$-secure encryption scheme (with universal ciphertexts), where $\mathcal{F}$ is the class of all $\mathbb{Z}_2$-linear functions and κ is the bit-length of an encryption key.

We now claim that such an encryption scheme exists assuming the nearly optimal hardness of $\textbf{Search-LWE}_{n,\ell,q,\chi_{\text{sec}},\chi_{\text{err}}}$ (Assumption 2.1) where q is even and $q \in \Omega(\lambda^2)$, χ_{sec} is the uniform distribution on $\{0,1\} \subseteq \mathbb{Z}_q$ (so the key length κ is n) and χ_{err} is the uniform distribution on $[-\frac{q}{4}, \frac{q}{4}) \subseteq \mathbb{Z}_q$.

Indeed, secret-key Regev encryption (Definition 2.17) with distributions $(\chi_{\text{sec}}, \chi_{\text{err}})$ as above immediately presents itself as a candidate encryption scheme. The reason that we choose χ_{sec} to be supported on $\{0,1\} \subseteq \mathbb{Z}_q$ is that $\mathcal{F}$-KDM security of this scheme for $\mathbb{Z}_2$-linear functions tightly follows from LWE. The folklore security reduction works as follows. Let $\varphi : \mathbb{F}_2^n \to \mathbb{F}_2^\ell$ be any affine function parameterized by a matrix C and vector $\mathbf{d}$ such that $\varphi(\mathbf{x}) = \mathbf{x}C+\mathbf{d}$. Given an LWE sample $(A, \mathbf{b} = \mathbf{s}A+\mathbf{e})$ with $A \leftarrow \mathbb{Z}_q^{n\times\ell}$ and $\mathbf{e}^t \leftarrow \chi_{\text{err}}$, one can efficiently produce a ciphertext, namely $(A-\frac{q}{2}\cdot C, \mathbf{b}+\frac{q}{2}\cdot\mathbf{d})$, that is *identically distributed* to a Regev encryption of $\varphi(\mathbf{s}) = \mathbf{s}C+\mathbf{d}$ (mod 2) with the above parameters. Therefore, if some adversary $\mathcal{A}$ when given a Regev encryption $\mathsf{Enc}(\mathbf{s}, \mathbf{s}B + \mathbf{c} \pmod 2)$ recovers $\mathbf{s}$ with probability ϵ, then the adversary $\mathcal{A}'$ that is given LWE samples $(A, \mathbf{b})$ and computes $\mathcal{A}(A - \frac{q}{2}C, \mathbf{b} + \frac{q}{2}\mathbf{d})$ as above will also recover $\mathbf{s}$ with probability ϵ.

Finally, we note that this scheme has universal ciphertexts (Definition 2.9) – indeed, for any $\mathbf{s}$, an encryption of a random bit-string under s is a uniformly random string – so this completes our first construction and security proof.

Arithmetic Randomized Encodings. We next generalize the construction from Section 2.4.3 to rely on forms of LWE with secrets that are *not* restricted to be elements of $\{0,1\}^n$, and thus more plausibly are nearly optimally secure. Specifically, letting the modulus $q = pq' \in \Omega(\lambda^2)$ be polynomially large, we will be able to have secrets that are uniformly random on the range $[-\frac{p}{2}, \frac{p}{2})^n$ and errors that are uniformly random in the range $[-\frac{q'}{2}, \frac{q'}{2})^\ell$, where p is prime, either $p \mid q'$ or $q' \geq p \cdot \lambda^2$. For example, setting $q' = p$, we could rely on an LWE assumption with secret and noise of order $\frac{1}{\sqrt{q}}$.

For this construction, we combine two tools: the KDM-secure encryption scheme of [ACPS09] (appropriately modified to have the desired statistical property) and a slightly non-standard variant of *arithmetic randomized encodings* over $\mathbb{Z}_p$ [AIK11]. We first describe the latter tool.

Theorem 2.27. *Let p be an arbitrary prime and let $\epsilon > 0$. Then, there is an unconditionally and information theoretically secure (non-universal) randomized encoding scheme $\mathsf{RE}_p^{\mathsf{approx}}$ for $\mathbb{Z}_p$-arithmetic circuits of depth at most d* that compute $\{0,1\}^\ell$-output functions *with the following properties:*

- $\mathsf{RE}_p^{\mathsf{approx}}$ *is perfectly secure.*
- $\mathsf{RE}_p^{\mathsf{approx}}$ *is $(1-\epsilon)$-approximately blind* ***for the output distribution that is uniform on*** $\{0,1\}^\ell \subseteq \mathbb{Z}_p^\ell$.
- *The size of a randomized encoding of (C, x) is $\mathsf{poly}(\log p, 2^d, |C|, \log(\frac{1}{\epsilon}))$.*
- *For any fixed choice of randomness r, the function $\mathsf{RE}_p^{\mathsf{approx}}.\mathsf{Enc}(C, x; r)$ is a $\mathbb{Z}_p$-affine function of (C, x).*

In order to prove Theorem 2.27, we first construct an intermediate randomized encoding using the techniques of [AIK11].

Theorem 2.28. *Let p be an arbitrary prime. Then, there is an unconditionally and information-theoretically secure (non-universal) randomized encoding scheme RE_p for $\mathbb{Z}_p$-arithmetic circuits of depth at most d with the following properties:*

- RE_p *is perfectly secure.*
- RE_p *is perfectly blind for the uniform distribution on $\mathbb{Z}_p^t$ (when the simulator is called on length-t outputs).*

- *The size of a randomized encoding of* (C, x) *is* $\mathsf{poly}(\log p, 2^d, |C|)$.
- *For any fixed choice of randomness* r*, the function* $\mathsf{RE}_p.\mathsf{Enc}(C, x; r)$ *is a* $\mathbb{Z}_p$*-affine function of* (C, x).

Proof. Our construction is a modification of [AIK11], Section 7.1; namely, we remove the key-shrinking gadget to obtain unconditional security.[22]

More formally, the construction is as follows: represent an arithmetic circuit

$$C = B_d \circ \ldots \circ B_2 \circ B_1$$

as a composition of d depth-1 circuits (with fan-in 2). We now inductively define encodings $\mathsf{Enc}_i(C, y^{(i)}; r^{(i)})$ and simulators $\mathsf{Sim}_i(C, y^{(d)})$ as follows:

- $\mathsf{Enc}_d(C, y^{(d)}; r) := y^{(d)}$ and $\mathsf{Sim}_d(C, y^{(d)}) = y^{(d)}$.
- For each $i < d$, define $f_i(C, y^{(i)}; r^{(i+1)}) = \mathsf{Enc}_{i+1}(C, B_{i+1}(y^{(i)}); r^{(i+1)})$. By the inductive hypothesis, each component $\mathbb{Z}_p$-element of $f_{i,\ell}(C, y^{(i)}; r^{(i+1)})$ is either a quadratic or a linear function of (two components of) $y^{(i)}$, with coefficients that may depend arbitrarily on $r^{(i+1)}$.
- For every linear component $f_i(\cdot)_\ell$ of the form $f_i(y^{(i)})_\ell = a_\ell \cdot (y_j^{(i)} + y_k^{(i)}) + b_\ell$, define

  $$\mathsf{Enc}_{i,\ell,0}(C, y^{(i)}; r^{(i+1)}||r) = a_\ell \cdot y_j^{(i)} + r$$

  $$\mathsf{Enc}_{i,\ell,1}(C, y^{(i)}, r^{(i+1)}||r) = a_\ell \cdot y_k^{(i)} + b_\ell - r,$$

  where $r \in \mathbb{Z}_p$ is uniformly random. Define corresponding simulators

  $$\mathsf{Sim}_{i,\ell,0}(C, \tilde{y}_\ell^{(i+1)}; r) = r$$

  $$\mathsf{Sim}_{i,\ell,1}(C, y_\ell^{(i+1)}; r) = \tilde{y}_\ell^{(i+1)} - r,$$

[22] [AIK11] notes that the construction with the key-shrinking gadget removed should give a randomized encoding scheme but does not actually analyze it. [AIK11] also notes that previous works give perfect randomized encodings with the parameters that we want, but it remains unclear if those schemes can be made perfectly blind.

where $r \in \mathbb{Z}_p$ is uniformly random.

- For every quadratic component $f_i(\cdot)_\ell$ of the form $a_\ell \cdot y_j^{(i)} \cdot y_k^{(i)} + b_\ell$, define

$$\mathsf{Enc}_{i,\ell,1,1}(C, y^{(i)}; r^{(i+1)}||r, s, t) = a_\ell \cdot y_j^{(i)} - r,$$

$$\mathsf{Enc}_{i,\ell,1,2}(C, y^{(i)}, r^{(i+1)}||r, s, t) = s \cdot a_\ell \cdot y_j^{(i)} + t,$$

$$\mathsf{Enc}_{i,\ell,2,1}(C, y^{(i)}, r^{(i+1)}||r, s, t) = y_k^{(i)} - s,$$

$$\mathsf{Enc}_{i,\ell,2,2}(C, y^{(i)}, r^{(i+1)}||r, s, t) = ry_k^{(i)} + b_\ell - t,$$

where r, s and t are uniformly random $\mathbb{Z}_p$-elements. Define corresponding simulators

$$\mathsf{Sim}_{i,\ell,1,1}(C, \tilde{y}_\ell^{(i+1)}; r, s, t) = r,$$

$$\mathsf{Sim}_{i,\ell,1,2}(C, \tilde{y}_\ell^{(i+1)}, r, s, t) = t,$$

$$\mathsf{Sim}_{i,\ell,2,1}(C, \tilde{y}_\ell^{(i+1)}, r, s, t) = s,$$

$$\mathsf{Sim}_{i,\ell,2,2}(C, \tilde{y}_\ell^{(i+1)}, r, s, t) = \tilde{y}_\ell^{(i+1)} - rs - t,$$

where r, s and t are uniformly random $\mathbb{Z}_p$-elements.

- Define the encoding algorithm

$$\mathsf{Enc}_i(C, y^{(i)}) = \left(\mathsf{Enc}_{i,\ell,b,c}(C, y^{(i)}; r^{(i+1)} \| r_\ell, s_\ell, t_\ell)\right)_{\ell,b,c}$$

and simulator

$$\mathsf{Sim}_i(C, y^{(d)}) = \left(\mathsf{Sim}_{i,\ell,b,c}(C, \tilde{y}_\ell^{(i+1)}; r_\ell \| s_\ell, t_\ell)\right)_{\ell,b,c}$$

where $\tilde{y}^{(i+1)} = (\tilde{y}_\ell^{(i+1)}) \leftarrow \mathsf{Sim}_{i+1}(C, y^{(d)})$.

Finally, the overall encoding algorithm $\mathsf{RE}_p.\mathsf{Enc}$ is defined to be Enc_0 with associated simulator $\mathsf{RE}_p.\mathsf{Sim} = \mathsf{Sim}_0$.

For decoding, $\mathsf{Dec}_d(\tilde{y}_d)$ is defined to output $\tilde{y}_d$, and Dec_i is defined to add every pair of "additive" encodings $(z_1, z_2) \mapsto z_1 + z_2$, combine multiplicative encodings by computing $(z_1, z_2, z_3, z_4) \mapsto z_1 z_3 + z_2 + z_4$, and then iteratively call Dec_{i+1} on the resulting concatenation of $\mathbb{Z}_p$-elements. The algorithm Dec_0 is then defined to be the decoding algorithm associated to $\mathsf{RE}_p.\mathsf{Enc}$.

Correctness of the above scheme is clear by inspection. We argue by induction that this scheme is perfectly private and perfectly blind.

Perfect blindness is shown inductively as follows: $\mathsf{Sim}_d(C, y^{(d)}) := y^{(d)}$ is a uniformly random string when $y^{(d)}$ is uniformly random. Moreover, if $\mathsf{Sim}_{i+1}(C, y^{(d)})$ is a uniformly random string when $y^{(d)}$ is a uniformly random string, then $\mathsf{Sim}_i(C, y^{(d)})$ is also uniformly random, as for each $\mathbb{Z}_p$-element $\tilde{y}_\ell^{(i+1)}$ of Sim_{i+1}, the four (or two, in the additive case) $\mathbb{Z}_p$-elements in the corresponding Sim_i-simulation are sampled to be uniformly random strings (r, s, t, u) subject to the equation $rs + t + u = \tilde{y}_\ell^{(i+1)}$ (or $r + s = \tilde{y}_\ell^{(i+1)}$ in the additive case). Thus, by induction we conclude that Sim_0 is perfectly blind.

Perfect privacy follows by a similar inductive argument; namely, $\mathsf{Sim}_d(C, y^{(d)})$ is clearly a perfectly private simulator for the identity function, and if $\mathsf{Sim}_{i+1}(C, y^{(d)})$ is a perfectly private simulator for the function $B_d \circ \ldots \circ B_{i+2}$, then we see that $\mathsf{Sim}_i(C, y^{(d)})$ is a perfectly private simulator for the function $B_d \circ \ldots \circ B_{i+1}$. To see this, we note that for every circuit-input pair $(C, y^{(i)})$, we have

$$\mathsf{Sim}_i\Big(C, y_d := (B_d \circ \ldots \circ B_{i+1})(y^{(i)})\Big) \equiv \Big(\mathsf{Sim}_{i,\ell,b,c}(C, \tilde{y}_\ell^{(i+1)}; r_\ell, s_\ell, t_\ell)\Big)_{\ell,b,c}$$

for $\tilde{y}^{(i+1)} \leftarrow \mathsf{Sim}_{i+1}(C, y^{(i)})$. By the induction hypothesis, we know that $\mathsf{Sim}_{i+1}(C, y_d)$ is identically distributed to $\tilde{y}^{(i+1)} \leftarrow \mathsf{Enc}_{i+1}\Big(C, B_{i+1}(y^{(i)})\Big)$. Thus, it suffices to show that the distribution $\Big(\mathsf{Sim}_{i,\ell,b,c}(C, \tilde{y}_\ell^{(i+1)}; r_\ell, s_\ell, t_\ell)\Big)_{\ell,b,c}$ is identical to the distribution $\Big(\mathsf{Enc}_{i,\ell,b,c}(C, y^{(i)}; r^{(i+1)}, (r_\ell, s_\ell, t_\ell))\Big)_{\ell,b,c}$. But for each ℓ, the corresponding component $\Big(\mathsf{Enc}_{i,\ell,b,c}(C, y^{(i)}; r^{(i+1)}, (r_\ell, s_\ell, t_\ell))\Big)_{b,c}$ is simply a uniformly random tuple $(\alpha, \beta, \gamma, \delta) \in \mathbb{Z}_p^4$ subject to the constraint that $\alpha\gamma + \beta + \delta = \tilde{y}_\ell^{(i+1)}$ (or a random tuple (α, β) subject to $\alpha + \beta = \tilde{y}_\ell^{(i+1)}$ in the additive case), which exactly matches the corresponding distribution $\left(\mathsf{Sim}_{i,\ell,b,c}(C, \tilde{y}_\ell^{(i+1)}; r_\ell, s_\ell, t_\ell)\right)_{b,c}$. This completes the induction, and hence the proof of Theorem 2.28. □

Using Theorem 2.28, we now prove Theorem 2.27.

Proof. The randomized encoding scheme $\mathsf{RE}_p^{\mathsf{approx}}$ for circuits of output length ℓ is defined as follows.

- $\mathsf{RE}_p^{\mathsf{approx}}.\mathsf{Enc}(C, x; \mathbf{R}, \mathbf{r}_0, \mathbf{r}_1)$ uses as randomness $\mathbf{R}$ for $\mathsf{RE}_p.\mathsf{Enc}$ as in Theorem 2.28 along with (for each $i \in [\ell]$) $\mathbb{Z}_p^{\log(\frac{\ell}{\epsilon})}$-elements[23] $r_{0,i}$ sampled uniformly from the set $\{0, 1, \ldots, \frac{p^{\log(\frac{\ell}{\epsilon})}-1}{2}\}$ and $\mathbb{Z}_p^{\log(\frac{\ell}{\epsilon})}$-elements $r_{1,i}$ sampled uniformly from the set $\{\frac{p^{\log(\frac{\ell}{\epsilon})}+1}{2}, \ldots, p^{\log(\frac{\ell}{\epsilon})} - 1\}$. It outputs

$$\mathsf{RE}_p.\mathsf{Enc}(C', (x, \mathbf{r}_0, \mathbf{r}_1); \mathbf{R})$$

  where $C'(x, \mathbf{r}_0, \mathbf{r}_1) = \mathbf{r}[y] := (r_{y_i,i})_i \in \mathbb{Z}_p^{\ell \log(\frac{\ell}{\epsilon})}$ for $y = C(x)$.

- $\mathsf{RE}_p^{\mathsf{approx}}.\mathsf{Dec}(C, \tilde{y})$ computes $\mathbf{r} = \mathsf{RE}_p.\mathsf{Dec}(C', \tilde{y})$ and then sets output bit y_i to 0 if and only if $0 \le r_i \le \frac{p^{\log(\frac{\ell}{\epsilon})}-1}{2}$ (and sets $y_i = 1$ otherwise).

- The simulator $\mathsf{RE}_p^{\mathsf{approx}}.\mathsf{Sim}(C, y)$ will sample $(\mathbf{r}_0, \mathbf{r}_1)$ as above and output $\mathsf{RE}_p.\mathsf{Sim}(C', \mathbf{r}_y)$.

Perfect correctness of the above scheme is clear by inspection. Moreover, perfect privacy is also clear: for any $(C, x, \mathbf{r}_0, \mathbf{r}_1)$, we know that $\mathsf{RE}_p.\mathsf{Sim}(C', \mathbf{r}[C(x)])$ is identical to the distribution $\mathsf{RE}_p.\mathsf{Enc}(C', (x, \mathbf{r}_0, \mathbf{r}_1))$, which immediately implies perfect privacy of the new scheme.

Finally, we see that the scheme is $(1-\epsilon)$-approximately blind, as for a uniformly random bit y_i, the resulting distribution on $r = r_{y_i}$ is a $(1 - \frac{1}{p^{\log(\frac{\ell}{\epsilon})}})$-multiplicative approximation of the uniform distribution on $\mathbb{Z}_p$ (and is $(1 - \frac{1}{p^{\log(\frac{\ell}{\epsilon})}})$-multiplicatively approximated by the same distribution). By repetition, we see that for a uniformly random y, the resulting distribution on $\mathbf{r} = \mathbf{r}[y]$ is $(1-\epsilon)$-multiplicatively comparable

[23] We interpret elements of $\mathbb{Z}_p^{\log(\frac{\ell}{\epsilon})}$ as represented by integers in the range $[0, p^{\log(\frac{\ell}{\epsilon})} - 1]$.

to the uniform distribution on $\mathbb{Z}_p^{\ell \log(\frac{\ell}{\epsilon})}$. Thus, $(1-\epsilon)$-approximate blindness follows from the perfect blindness of RE_p. $\square$

We now combine Theorem 2.27 with Imported Theorem 2.26 to obtain a randomized encoding scheme $\widetilde{\mathsf{RE}}_p$ satisfying the structural and security properties required to be used with an [ACPS09]-like encryption scheme. In this scheme, we consider the following notion of evaluating boolean circuits on $\mathbb{Z}_p$-inputs: if C is a boolean circuit with input length $\kappa \cdot \lceil \log(p) \rceil$ and $x \in \mathbb{Z}_p^\kappa$, we define $C(x) := C(\llbracket x \rrbracket := (\llbracket x \rrbracket_1, \ldots, \llbracket x \rrbracket_{\kappa \lceil \log(p) \rceil}))$, where $\llbracket x \rrbracket_i$ is defined to be the ith bit of x in the representation $[0, p-1]^\kappa \subseteq (\{0,1\}^{\lceil \log(p) \rceil})^\kappa$. We consider randomized encodings of circuit-input pairs (C, x) of this form, in which encodings are strings over the alphabet $\mathbb{Z}_p$.

Theorem 2.29. *Let $p = p(\kappa)$ be an arbitrary prime (sequence) and $\epsilon = \epsilon(\kappa) > 0$. If sub-exponentially secure one-way functions exist, there is a universal randomized encoding scheme $\widetilde{\mathsf{RE}}_p$ for polynomial-size boolean circuits with $\mathbb{Z}_p$-inputs with the following properties:*

- *$\widetilde{\mathsf{RE}}_p$ is $o(2^{-\kappa})$-secure, and all operations run in time $\mathsf{poly}(\kappa, p)$.*
- *$\widetilde{\mathsf{RE}}_p$ is $(1-\epsilon)$-approximately blind.*
- *For any fixed choice of randomness r and circuit C, the function $\widetilde{\mathsf{RE}}_p.\mathsf{Enc}(C, x; r)$ is a $\mathbb{Z}_p$-affine function of x.*

Proof. The randomized encoding scheme $\widetilde{\mathsf{RE}}_p$ is a certain kind of composition of RE_p with point-and-permute garbled circuits (which we denote by RE).[24] More specifially, $\widetilde{\mathsf{RE}}_p$ works as follows:

- Input: A circuit C, input $x \in \mathbb{Z}_p^n$, and randomness r_1, r_2.
- Compute $f_2(r_1)$, where $\mathsf{RE}.\mathsf{Enc}(C, \llbracket x \rrbracket; r_1) = f_1(C, \llbracket x \rrbracket; r_1) || f_2(r_1)$ as in Imported Theorem 2.26.

[24] This does not exactly match the usual notion of composition, as in [AIK04] Lemma 4.11.

- Output $\mathsf{RE}_p.\mathsf{Enc}(\tilde{f}_1, (C, x, r_1, f_2(r_1); r_2)$, where $\tilde{f}_1(C, x, r, r') \overset{\mathsf{def}}{=} (f_1(C, [\![x]\!]; r), r')$ is interpreted as a $\mathbb{Z}_p$-arithmetic circuit and the bit-strings C and r are interpreted as strings over the alphabet $\{0,1\} \subseteq \mathbb{Z}_p$.

To see that this scheme is efficient, we note that RE_p is only used to compute randomized encodings of a function $\tilde{f}_1(C, x, r, r')$ with the property that each output bit depends on a constant number of bits of $[\![x]\!]$ and a constant number of bits of (C, r, r'). This in turn depends on only a constant number of $\mathbb{Z}_p$-blocks of the input (C, x, r, r'). We claim that any such function can be computed by a $O(\log(p))$-depth $\mathbb{Z}_p$-arithmetic circuit: a function $\tilde{f}(z_1, \ldots, z_c)$ of c-many $\mathbb{Z}_p$ symbols can be expressed in the following form:

$$\tilde{f}(z_1, \ldots, z_c) = \sum_{a_1, \ldots, a_c \in \mathbb{Z}_p} \tilde{f}(a_1, \ldots, a_c) \prod_{i=1}^{c} (1 - (z_i - a_i)^{p-1}).$$

The outer sum can be computed in $\log(p)$ depth, and each term can be computed in at most $1 + \log(c) + \log(p)$ depth with repeated squaring. Thus, RE_p can be used to encode the function $\tilde{f}$ with the desired efficiency.

The simulator for this scheme $\widetilde{\mathsf{RE}}_p.\mathsf{Sim}(y)$ will simply call $\mathsf{RE}_p.\mathsf{Sim}(\mathsf{RE}.\mathsf{Sim}(y))$. Simulation security follows from a standard hybrid argument.

Finally, $(1 - \epsilon)$-approximate blindness follows because $\mathsf{RE}.\mathsf{Sim}(U_\ell)$ is identical to the uniform distribution on binary strings of the appropriate length by the perfect blindness of RE, and so $\widetilde{\mathsf{RE}}_p.\mathsf{Sim}(\mathsf{RE}.\mathsf{Sim}(U_\ell))$ is $(1 - \epsilon)$-approximately comparable to the uniform distribution on $\mathbb{Z}_p$-strings of the appropriate length by the $(1 - \epsilon)$-approximate blindness of $\widetilde{\mathsf{RE}}_p$. □

A Scheme from Exponential LWE with Moderately Small Secrets. Combining Theorem 2.25 with Theorem 2.29, we conclude that to construct a $\mathsf{SIZE}(\kappa^c)$-KDM $2^{-\kappa}\mathsf{poly}(\kappa)$-secure encryption scheme with universal ciphertexts, it suffices to construct a $\mathcal{F}$-KDM $2^{-\kappa}\mathsf{poly}(\kappa)$-secure encryption scheme (with universal ciphertexts), where $\mathcal{F}$ is the class of all $\mathbb{Z}_p$-linear functions.

We now claim that such an encryption scheme exists assuming the exponential hardness of $\mathsf{LWE}_{n,\ell,q,\chi_{\text{sec}},\chi_{\text{err}}}$ (Assumption 2.1) where the modulus $q = pq' \in \Omega(\lambda^2)$ is polynomially large, χ_{sec} is the uniform distribution on $[-\frac{p}{2}, \frac{p}{2}) \subseteq \mathbb{Z}_q$ and χ_{err} is the uniform distribution on $[-\frac{q}{2p}, \frac{q}{2p}) \subseteq \mathbb{Z}_q$. In order for this assumption to be plausible, we pick q' such that either $q' \geq p \cdot \lambda^2$ or q' is divisible by p.

To do this, we will use a modification of secret-key Regev encryption in the spirit of [ACPS09]. Our scheme $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ is as follows.

- $\mathsf{Gen}(1^n, q, p)$ samples a uniformly random $s \leftarrow [-\frac{p}{2}, \frac{p}{2})^n \subseteq \mathbb{Z}_q^n$.
- $\mathsf{Enc}(s, m \in \mathbb{Z}_p^\ell)$ samples a uniformly random matrix $A \leftarrow \mathbb{Z}_q^{n\times\ell}$ and error $e \leftarrow [-\frac{q}{2p}, \frac{q}{2p})^\ell$ and outputs $(A, s^tA + e^t + q' \cdot m)$.
- $\mathsf{Dec}(s, \mathsf{ct})$ interprets $\mathsf{ct} = (A, b)$, computes $b - s^tA \pmod q$, rounds each entry to the nearest multiple of q', and divides each entry by q'.

Correctness of the encryption scheme is clear. Moreover, $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ satisfies the statistical property required of a CCRR-compatible encryption scheme, as for any fixed s, a random encryption $\mathsf{Enc}(s, U_{\ell,p})$ is identical to a uniformly random element of $\mathbb{Z}_q^{n\times\ell} \times \mathbb{Z}_q^\ell$.

Finally, we see that our scheme satisfies exponential KDM-security for $\mathbb{F}_p$-affine functions of the secret key by a similar reduction to that of Section 2.4.3. Namely, for a secret $s \leftarrow [-\frac{p}{2}, \frac{p}{2})^n$, given an $\mathsf{LWE}_{n,\ell,q,\chi_{\text{sec}},\chi_{\text{err}}}$ sample $(A, b = s^tA + e^t)$ with $A \leftarrow \mathbb{Z}_q^{n\times\ell'}$, one can efficiently produce a ciphertext $(A - 2q'B, b + c)$ that is *identically distributed* to a Regev encryption of $s^tB + c \pmod p$ with the above parameters. Therefore, if some adversary $\mathcal{A}$ that is given a Regev encryption $\mathsf{Enc}(s, s^tB + c \pmod p)$ recovers s with probability ϵ, then the adversary $\mathcal{A}'$ that is given an LWE sample (A, b) and computes $\mathcal{A}(A - 2q'B, b + c)$ as above will also recover s with probability ϵ.

Discussion. *Our scheme most notably differs from that of [ACPS09] in our choice of error distribution (which is also made possible by the fact that we consider a secret-key variant). Namely, [ACPS09] takes the error distribution χ_{err} to be the same as χ_{sec} (and they use Gaussian distributions of width $\Theta(p)$ rather than uniform distributions as well). Using uniformly random secrets (over intervals) and errors is required for the* *exponential security and statistical properties of our encryption scheme to plausibly hold. However, we note that it is also possible to rely on an* LWE *assumption in which our error distribution* χ_{err} *is instead uniform on* $[-\frac{p}{2}, \frac{p}{2})$ *(i.e. the same as* χ_{sec}*). Namely, this* LWE *variant actually follows from the* LWE *variant that we assume here, with the caveat that we must then take* $q > \ell \cdot p$*. The reduction is similar to the high-noise-to-low-noise reduction in Theorem 2.20.*

A Scheme from ElGamal Encryption. In addition to our LWE-based constructions, we note that by combining our amplification theorem (Theorem 2.25) with point-and-permute garbled circuits, we can generically reduce the problem of constructing $\mathsf{SIZE}(\kappa^c)$-KDM secure encryption schemes (with universal ciphertexts and almost optimal security) to constructing *circular* secure encryption schemes (with the same properties). In particular, we can plug in the variant of ElGamal encryption defined in [CCRR18].[25] We immediately conclude that if this variant of ElGamal encryption satisfies almost optimal circular security, then NIZK arguments exist (combining Theorem 2.25 and Theorem 2.52).

2.5 Round-by-Round Soundness and Fiat-Shamir

In this section we define the notion of an interactive proof with *round-by-round* soundness, and prove that correlation intractability for a specific related relation is sufficient for a hash family to ensure that the associated Fiat-Shamir heuristic is sound.

2.5.1 Definitions: Interactive Proofs and Arguments

We being by recalling the definitions of interactive proofs and arguments. We focus on *doubly-efficient* proof-systems, in which the prover is polynomial-time and the verifier is quasi-linear.

[25] In [CCRR18], this scheme was assumed to satisfy almost optimal KDM-security for arbitrary KDM functions.

Definition 2.30. *A* doubly-efficient interactive proof *(resp.,* interactive argument*) for a promise problem $\mathcal{L} = (\mathcal{L}_{\mathsf{yes}}, \mathcal{L}_{\mathsf{no}})$ is a pair (P, V) of interactive algorithms satisfying:*

- **Completeness.** *For any $x \in \mathcal{L}_{\mathsf{yes}}$, when P and V interact on common input x, the verifier V outputs 1 with probability 1.*
- **Soundness.** *For any $x \in \mathcal{L}_{\mathsf{no}} \cap \{0,1\}^n$ and any* unbounded *(resp.,* polynomial-time*) interactive P^*, when P^* and $V(x)$ interact, the probability that V outputs 1 is a negligible function of n.*
- **Efficiency.** *V runs in time $\tilde{O}(n)$ and P runs in $\mathsf{poly}(n)$ time, where n is the input length.*

The protocol is public coin *if each of V's messages is an independent uniformly random string of some length (and the verifier's decision to accept or reject does not use any secret state).*

Definition 2.31. *A* two-message *argument scheme is one in which the interaction consists of a single message from the verifier to the prover followed by a single message from the prover to the verifier. The scheme is* delayed input *if the joint distribution of the first message* together with the resulting verifier state *also depends only on n.*

A delayed-input two-message argument scheme is said to be adaptively sound *if soundness holds for a cheating prover that chooses x* after *seeing the verifier's first message. The scheme is* publicly verifiable *if the verifier's first message includes the verifier's subsequent state.*

2.5.2 Round-by-Round Soundness

Definition 2.32 (Round-by-Round Soundness)**.** *Let $\Pi = (P, V)$ be a $2r$-message public coin interactive proof system for a language L. For any $x \in \{0,1\}^*$, and any prefix τ of a protocol transcript, let $V(x, \tau)$ denote the distribution of the next message (or output) of V when the transcript so far is τ and V was executed on input x.*

We say that Π has round-by-round soundness error $\epsilon(\cdot)$ *if there exists a deterministic (not necessarily efficiently computable) function* State *that takes as input an instance*

x and a transcript prefix τ and outputs either acc *or* rej *such that the following properties hold:*

1. *If $x \notin L$, then $\mathsf{State}(x, \emptyset) = \mathsf{rej}$, where $\emptyset$ denotes the empty transcript.*

2. *If $\mathsf{State}(x, \tau) = \mathsf{rej}$ for a transcript prefix τ, then for every potential prover message α, it holds that*

$$\Pr_{\beta \leftarrow V(x, \tau|\alpha)} \Big[\mathsf{State}\Big(x, \tau|\alpha|\beta\Big) = \mathsf{acc}\Big] \leq \epsilon(n)$$

3. *For any* full[26] *transcript τ, if $\mathsf{State}(x, \tau) = \mathsf{rej}$ then $V(x, \tau) = 0$.*

We say that Π is round-by-round sound *if it has round-by-round soundness error ϵ for some $\epsilon(n) = \mathrm{negl}(n)$.*

Remark 2.33. *The completeness condition of the interactive proof implies that for $x \in L$ (i.e., a YES instance) and an honestly generated transcript τ, with high probability over the coins tossed, it holds that $\mathsf{State}(x, \tau) = \mathsf{acc}$.*

Before diving into the proof that the Fiat-Shamir paradigm can be applied to any interactive proof with round-by-round soundness (in Section 2.5.3), we first discuss some basic properties of these type of protocols.

Round-by-round Soundness vs. Standard Soundness. A first basic observation is that round-by-round soundness implies standard soundness (with a loss proportional to the number of rounds).

Proposition 2.34. *Let Π be $2r$-message interactive proof with round-by-round soundness error ϵ. Then, Π has standard soundness error $r \cdot \epsilon$.*

Proof. By a union bound over the error in all of the rounds. □

Conversely, standard soundness implies *some* (smaller) amount of round-by-round soundness.

[26] By a full transcript, we mean a transcript for which the verifier halts.

Proposition 2.35. *Let Π be a $2r$-message interactive proof with soundness error μ. Then, Π has round-by-round soundness error $\mu^{\frac{1}{r}}$.*

Proof. Let $\Pi = (P, V)$ denote a $2r$-message (public coin) interactive proof with soundness error μ. We associate to Π the following State function, defined inductively for partial transcripts of length $2i$.

- Given a full transcript τ, we define $\mathsf{State}(x, \tau) = \mathsf{acc}$ if and only if $V(x, \tau)$ accepts.
- Inductively, given a transcript τ of length $2i$, we define $\mathsf{State}(x, \tau) = \mathsf{acc}$ if and only if there exists a message α^*_{i+1} such that

$$\Pr_{\beta_{i+1}}\left[\mathsf{State}(x, \tau|\alpha^*_{i+1}|\beta_{i+1}) = \mathsf{acc}\right] > \mu^{\frac{1}{r}}.$$

We claim that Π has round-by-round soundness error $\mu^{\frac{1}{r}}$ with respect to this State function. We note that properties (2) and (3) of round-by-round soundness are satisfied by construction. All that we need to verify is property (1), i.e., that $\mathsf{State}(x, \emptyset) = \mathsf{rej}$ for $x \notin L$. To see this, we note that if $x \notin L$ but $\mathsf{State}(x, \emptyset) = \mathsf{acc}$, then by definition of State, there exists a prover strategy P^* such that

$$\Pr_{\beta=(\beta_1,\ldots,\beta_r)}[\mathsf{State}(x, \tau_{P^*,\beta}) = \mathsf{acc}] > (\mu^{\frac{1}{r}})^r = \mu,$$

where $\tau_{P^*,\beta}$ denotes the transcript associated to prover strategy P^* and verifier messages β. This contradicts the μ-soundness of Π (since if $\mathsf{State}(x, \tau_{P^*,\beta}) = \mathsf{acc}$ then the verifier accepts). Thus, we conclude that Π satisfies round-by-round $\mu^{\frac{1}{r}}$-soundness with respect to State, as desired. □

Finally, we note that Proposition 2.35 is tight in its security loss.

Proposition 2.36. *There exists an r-round interactive proof with soundness error 2^{-r} that does not have round-by-round soundness error $\frac{1}{2} - \epsilon$ for any $\epsilon > 0$.*

Proof. Consider the following interactive proof for the empty language. On input $x \in \{0,1\}^n$, the protocol proceeds as follows. In each round the prover sends nothing, then the verifier tosses a fresh coin and sends the result to the prover. After r rounds the verifier accepts if and only if all coin tosses were 0.

Clearly this constitutes an interactive proof for the empty language (with soundness error 2^{-r}). Suppose that the protocol has round-by-round soundness error $1/2-\epsilon$ and let State be a corresponding state function. Fix also an arbitrary input x^* (a NO input, needless to say).

By the first property of round-by-round soundness $\mathsf{State}(x^*, \emptyset) = \mathsf{rej}$. On the other hand, by the third property, it holds that $\mathsf{State}(x^*, 0^r) = \mathsf{acc}$ (since the verifier accepts in case all coin tosses were 0).

Thus, there must exist $i \in [r]$ such that $\mathsf{State}(x^*, 0^i) = \mathsf{rej}$ and $\mathsf{State}(x^*, 0^{i+1}) = \mathsf{acc}$. This means that

$$\Pr_{b \in \{0,1\}} \left[\mathsf{State}(x^*, 0^i | b) = \mathsf{acc} \right] \geq \frac{1}{2},$$

in contradiction to the second property of round-by-round soundness. □

Parallel Repetition and Round-by-Round Soundness. Given an interactive proof $\Pi = (P, V)$ we can consider the k-fold parallel repetition of Π, denoted by $\Pi^k = (P^k, V^k)$, in which (P, V) is executed k times independently and the verifier accepts if and only if a majority of executions accept.[27] It is known that parallel repetition reduces the completeness error and soundness error of interactive proofs at an exponential rate (see [Gol99, Lemma C.1]).[28] Together with Proposition 2.35, this implies that *any* sound public coin proof system can be converted into one satisfying round-by-round soundness.

Corollary 2.37. *Suppose that* Π *is a* $2r$*-round (public coin) proof system with soundness error* μ*. Then,* Π^k *has* round-by-round *soundness error* $\mu^{\frac{k}{r}}$*.*

[27] In case the base protocol (P, V) has perfect completeness, it suffices for V^k to check that *all* executions accept.

[28] The fact that the completeness error is reduced at an exponential rate is trivial. Soundness is more difficult to analyze though since a cheating prover for V^k does not have to act independently on the k executions. Nevertheless, it was shown [Gol99, Lemma C.1] that the soundness error is reduced at an exponential rate.

2.5.3 Round-by-Round Soundness and Fiat-Shamir

The main result of this section is that the Fiat-Shamir transform for compressing a public-coin interactive proof Π into a non-interactive transform is provably (adaptively) sound when applied to *round-by-round sound* interactive proofs using a hash family satisfying a restricted form of correlation intractability.

Specifically, we show that it suffices for the hash family to be correlation intractable with respect to a specific relation, which we now define. Let Π be an interactive proof with round-by-round soundness error ϵ and let State be a corresponding state function. For every $n \in \mathbb{N}$, we define a relation $R^{(n)}_{\mathsf{State}}$ as follows:

$$R^{(n)}_{\mathsf{State}} \overset{\mathsf{def}}{=} \left\{ \Big(\big(x, \tau | \alpha \big), \beta \Big) : \begin{array}{c} x \in \{0,1\}^n, \\ \mathsf{State}(x,\tau) = \mathsf{rej} \\ \text{and} \\ \mathsf{State}(x, \tau|\alpha|\beta) = \mathsf{acc} \end{array} \right\}.$$

We define the relation ensemble $R_{\mathsf{State}} = (R^{(n)}_{\mathsf{State}})_{n \in \mathbb{N}}$.

Note that R_{State} is ϵ-sparse, since Π has round-by-round soundness ϵ. When there is a canonical choice of the function State for a protocol Π, we will often write R_Π to denote R_{State}.

Theorem 2.38. *Suppose that $\Pi = (P, V)$ is a $2r$-message public-coin interactive proof for a language L with perfect completeness, $\mathsf{polylog}(n)$ total bits of prover-to-verifier communication, and round-by-round soundness with a corresponding state function State. Let X_n denote the set of partial transcripts (including the input and all messages sent) and let Y_n denote the set of verifier messages when Π is executed on an input of length n. If a hash family $\mathcal{H} = \{\mathcal{H}_n : X_n \to Y_n\}$ is R_{State}-correlation intractable and evaluable in time $\tilde{O}(n)$,[29] then the algorithms $(\mathsf{Gen}, \tilde{P}, \tilde{V})$ as defined below constitute an adaptively sound publicly verifiable argument for L.*

- *On input 1^n, Gen samples $H \leftarrow \mathcal{H}_n$, and publishes H as a common reference*

[29] This is only due to our definition of a doubly-efficient argument, which stringently requires that the verifier's running time is $\tilde{O}(n)$.

string (or common random *string if $H \leftarrow \mathcal{H}_n$ is a uniformly random binary string of some length.*

- *On input x, the prover $\tilde{P}$ sends the r strings $\alpha_1, \ldots, \alpha_r$ that P would send on input x if the verifier's messages were given by $\beta_j = H(x, \alpha_1|\beta_1|\cdots|\alpha_j)$ for $j \in [r]$*

- *The verifier $\tilde{V}$, on input x^* and $\alpha_1^*, \ldots, \alpha_r^*$ (which might be chosen maliciously) iteratively computes*

$$\beta_j^* = H\Big(x^*, \alpha_1^*|\beta_1^*|\alpha_2^*|\ldots|\beta_{j-1}^*|\alpha_j^*\Big)$$

  for each $j \in [r]$. The verifier then accepts if and only if $V(x^*, \alpha_1^*|\beta_1^*|\ldots|\alpha_r^*|\beta_r^*) = 1$.

Remark 2.39 (On Interactive Proofs with Imperfect Completeness)**.** *Theorem 2.38 applies to protocols with* perfect *completeness. However, it can be easily extended to protocols with imperfect completeness by further requiring that the correlation intractable hash function is r-wise independent (so as to assure the correct distribution of verifier messages). This can be done without loss of generality by xor-ing the (bounded) correlation intractable hash function with an r-wise independent hash function, which preserves (bounded) correlation intractability.*

Proof of Theorem 2.38. Completeness follows immediately from the perfect completeness of (P, V).

We proceed to show the adaptive soundness of the argument scheme. Suppose that a cheating prover P^* given input $(1^n, H)$ produces, with probability at least $\epsilon = \epsilon(n)$, a string $x^* \in \{0,1\}^n \setminus L$ and $(\alpha_1^*, \ldots, \alpha_r^*)$ such that V accepts the transcript derived from $H(\cdot)$. Let τ_i denote the transcript prefix $\alpha_1^*|\beta_1^*|\cdots\alpha_i^*|\beta_i^*$ with β_j^* defined as above.

Properties 1 and 3 of round-by-round soundness (see Definition 2.32) imply that for any accepting transcript τ for $x \notin L$ there is at least one index $i \in [r]$ such that $\mathsf{State}(x, \tau_i) = \mathsf{rej}$ and $\mathsf{State}(x, \tau_{i+1}) = \mathsf{acc}$. Thus, there must exist some index $i_n^* \in [r]$

such that with probability at least $\frac{\epsilon}{r}$, the output of P^* satisfies that $\mathsf{State}(x, \tau_{i^*}) = \mathsf{rej}$ and $\mathsf{State}(x, \tau_{i^*+1}) = \mathsf{acc}$.

This fact can be used to construct an adversary $\mathcal{A} = \{\mathcal{A}_\lambda\}$ that violates the R_{State}-correlation intractability of $\mathcal{H}$: on input $H \leftarrow \mathcal{H}_n$, $\mathcal{A}_\lambda$ runs $P^*(1^n, H)$ to obtain x^* and $(\alpha_1^*, \ldots, \alpha_r^*)$, computes $\beta_j = H(x^*, \tau_{j-1}|\alpha_j)$ for all j, and outputs $\tau_{i^*}|\alpha_{i+1}^*$. This is a contradiction, so the protocol must be adaptively sound. □

2.6 Publicly Verifiable SNARG

We present our construction of a publicly verifiable SNARG based on the GKR interactive protocol. We begin in Section 2.6.1 by recalling some standard algebraic facts and notations.

Font Conventions. Throughout this section we will use the convention that blackboard bold lowercase (e.g., $\mathbb{z}$) is used for field elements whereas standard bold lowercase (e.g., $\mathbf{z}$) is used for bits. Likewise, we use $\bar{\mathbb{z}}$ to denote vectors of field elements and $\bar{\mathbf{z}}$ to denote bit strings.

2.6.1 Fields and Polynomials

We recall the definition of the multilinear extension and explicit representations of finite fields.

Definition 2.40 (Multilinear Extension)**.** *For any function $f : \{0,1\}^n \to \{0,1\}$ and any field $\mathbb{F}$, the* multilinear extension of f over $\mathbb{F}$ *is the (uniquely) defined multilinear polynomial $\hat{f} : \mathbb{F}^n \to \mathbb{F}$ satisfying $\hat{f}(x) = f(x)$ for each $x \in \{0,1\}^n$.*

The polynomial $\hat{f}(\mathbb{z})$ is given explicitly by the formula

$$\hat{f}(\mathbb{z}) = \sum_{x \in \{0,1\}^n} f(x) \cdot \beta_{x \to \bar{\mathbb{z}}}$$

where $\beta_{x \to \bar{\mathbb{z}}} \overset{\mathsf{def}}{=} \prod_{i \in [n]} \Big(x_i \cdot \mathbb{z}_i + (1 - x_i) \cdot (1 - \mathbb{z}_i) \Big)$.

When the field $\mathbb{F}$ is clear from the context, we will omit it and simply say that $\hat{f}$ is the multilinear extension of f.

Definition 2.41. *A $T_{\mathbb{F}}(\cdot)$-time* explicit representation *of a finite field ensemble $\mathbb{F} = \{\mathbb{F}_i\}_{i \in \mathcal{I}}$ is an algorithm for solving each of the following problems in time $T_{\mathbb{F}}(i)$ given an index $i \in \mathcal{I}$.*

- **Field Membership.** *Given an additional string z, decide whether or not $z \in \mathbb{F}_i$.*
- **Enumerability.** *Evaluate some bijection $\varphi_i : [|\mathbb{F}_i|] \to \mathbb{F}_i$.*
- **Explicit 0 and 1.** *Compute $0 \in \mathbb{F}_i$ and $1 \in \mathbb{F}_i$.*
- **Efficient Field Operations.** *Evaluate the operations $+$, $-$, $\times$, and $\div$ on $\mathbb{F}_i$.*
- **Sampleable.** *Sample from the uniform distribution on $\mathbb{F}_i$.*

When a $T_{\mathbb{F}}(\cdot)$-time explicit representation exists, we say that $\mathbb{F}$ is $T_{\mathbb{F}}(\cdot)$-time representable.

2.6.2 GKR: Round by Round Soundness and Efficient Sampleability

In this section, we briefly describe the interactive proof system of Goldwasser, Kalai, and Rothblum [GKR08], hereafter referred to as GKR. We explain why GKR (or rather a simplification due to Goldreich [Gol17]) has round-by-round soundness, and we show that the corresponding relation (as defined in Section 2.5) can be sampled in polynomial time.

We start by using a result from [Gol17] that allows one to transform uniform low depth circuits into a form that is convenient for the GKR protocol.

Imported Lemma 2.42 ([Gol17])**.** *If $\mathcal{L}$ is a promise problem decidable by an ensemble of log-space uniform boolean circuits of size $S' = S'(n)$ (without loss of generality $S'(n) \geq n$) and depth $d' = d'(n)$, then $\mathcal{L}$ is also decidable by an ensemble $\{C_n\}$ of boolean circuits that satisfies the following uniformity properties:*

- *C_n has size $S(n) \leq \mathsf{poly}(S'(n))$ and depth $d(n) \leq d'(n) \cdot \mathsf{polylog}(S'(n))$. Assume without loss of generality that $S(n)$ is a power of two, and define $s(n) \overset{\mathsf{def}}{=} \log_2 S(n)$.*

- *The gates of C_n have fan-in 2, and each compute either $\oplus$ or $\wedge$.*

- *The gates of C_n can be (uniquely) partitioned into layers such that the inputs to a gate in layer i are outputs of gates in layer $i-1$, with the input wires viewed as layer 0.*

- *The wires of C_n can be labeled with the numbers 1 through $S(n)$ (equivalently with $s(n)$-bit strings) so that:*

  - *The first n wires of C_n are the input wires.*

  - *The last wire of C_n is the output wire.*

  - *Let "wiring predicates" $\mathsf{add}_n, \mathsf{mult}_n : \left(\{0,1\}^{s(n)}\right)^3 \to \{0,1\}$ be defined so that add_n (respectively, mult_n) applied to (w_1, w_2, w_3) is 1 iff w_3 is an $\oplus$ (respectively, $\wedge$) gate whose input wires are w_1 and w_2, in that order.*

    Then both add_n and mult_n are computable by $\mathsf{polylog}(n)$-sized boolean formulas that themselves are computable from n in $\mathsf{polylog}(n)$ time. In particular this implies that over any $T_{\mathbb{F}}(\cdot)$-time representable finite field ensemble $\mathbb{F} = \{\mathbb{F}_i\}$, there exist $\mathsf{polylog}(n)$-degree extensions $\widetilde{\mathsf{add}}_{n,i}, \widetilde{\mathsf{mult}}_{n,i} : \left(\mathbb{F}_i^{s(n)}\right)^3 \to \mathbb{F}_i$ that are evaluable in time $\mathsf{polylog}(n) \cdot T_{\mathbb{F}}(i)$.

Low-Degree Arithmetization. GKR depends on several polynomials, which we now define. Fix $\mathcal{L} = (\mathcal{L}_{\mathsf{yes}}, \mathcal{L}_{\mathsf{no}})$ to be any promise problem that is decidable by log-space uniform circuits of size $S'(n)$ and depth $d'(n)$.[30] Let $\{C_n\}$ denote a circuit family that decides $\mathcal{L}$ as in the conclusion of Imported Lemma 2.42.

For any $x \in \{0,1\}^n$, any field $\mathbb{F}$, any $i \in \{0, \ldots, d(n)\}$, and any $j \in [3 \cdot s(n)]$, we define polynomials $\hat{V}_{x,\mathbb{F}}^{(i)} : \mathbb{F}^{s(n)} \to \mathbb{F}$ and $P_{x,j,\mathbb{F}}^{(i)} : \mathbb{F}^j \times \mathbb{F}^{s(n)} \to \mathbb{F}$ as follows.

[30] Recall that log-space uniformity implies that $S'(n) = \mathsf{poly}(n)$.

We first define a function $V_x^{(i)} : \{0,1\}^{s(n)} \to \{0,1\}$ so that $V_x^{(i)}(w)$ is 1 iff wire w is in layer i and carries the value 1 when C_n is evaluated on x. The polynomial $\hat{V}_{x,\mathbb{F}}^{(i)}$ is defined as the multi-linear extension of $V_x^{(i)}$ over the field $\mathbb{F}$ (see Definition 2.40 for the definition of the multilinear extension).

The polynomial $P_{x,3s(n),\mathbb{F}}^{(i)} : \mathbb{F}^{4s(n)} \to \mathbb{F}$ is defined as

$$P_{x,3s(n),\mathbb{F}}^{(i)}(\bar{\mathbb{w}}_1, \bar{\mathbb{w}}_2, \bar{\mathbb{w}}_3, \bar{\mathbb{w}}) \stackrel{\mathsf{def}}{=} \left(\begin{array}{c} \widetilde{\mathsf{add}}_n(\bar{\mathbb{w}}_1, \bar{\mathbb{w}}_2, \bar{\mathbb{w}}_3) \cdot \left(\hat{V}_x^{(i-1)}(\bar{\mathbb{w}}_1) + \hat{V}_x^{(i-1)}(\bar{\mathbb{w}}_2) \right) \\ + \\ \widetilde{\mathsf{mult}}_n(\bar{\mathbb{w}}_1, \bar{\mathbb{w}}_2, \bar{\mathbb{w}}_3) \cdot \hat{V}_x^{(i-1)}(\bar{\mathbb{w}}_1) \cdot \hat{V}_x^{(i-1)}(\bar{\mathbb{w}}_2) \end{array} \right) \cdot \beta_{\bar{\mathbb{w}}_3 \to \bar{\mathbb{w}}}. \tag{2.2}$$

with $\beta_{\bar{\mathbb{w}}_3 \to \bar{\mathbb{w}}}$ as in Definition 2.40 on page 98. For $j \in \{0, \ldots, 3s(n)-1\}$, the we define a polynomial $P_{x,j,\mathbb{F}}^{(i)}$ as follows

$$P_{x,j,\mathbb{F}}^{(i)}(\mathbb{z}_1, \ldots, \mathbb{z}_j, \bar{\mathbb{w}}) \stackrel{\mathsf{def}}{=} \sum_{z_{j+1} \in \{0,1\}} P_{x,j+1,\mathbb{F}}^{(i)}(\mathbb{z}_1, \ldots, \mathbb{z}_j, z_{j+1}, \bar{\mathbb{w}}) \tag{2.3}$$

for $\mathbb{z}_1, \ldots, \mathbb{z}_j \in \mathbb{F}$ and $\bar{\mathbb{w}} \in \mathbb{F}^{s(n)}$. The polynomials $P_{x,j,\mathbb{F}}^{(i)}$ are often referred to as the "sumcheck polynomials", arising from the sumcheck protocol of [LFKN90] that we are implicitly using.

By the definitions of the wiring predicates and multi-linear extension, it holds for any i, any $\bar{\mathbb{w}}$, and any field $\mathbb{F}$ of characteristic two that

$$\begin{aligned} \hat{V}_{x,\mathbb{F}}^{(i)}(\bar{\mathbb{w}}) &= P_{x,0,\mathbb{F}}^{(i)}(\bar{\mathbb{w}}) \\ &= \sum_{\bar{w}_1, \bar{w}_2, \bar{w}_3 \in \{0,1\}^{s(n)}} P_{x,3s(n),\mathbb{F}}^{(i)}(\bar{w}_1, \bar{w}_2, \bar{w}_3, \bar{\mathbb{w}}) \end{aligned} \tag{2.4}$$

These polynomials each have degree $\mathsf{polylog}(n)$, and the relations between them are at the heart of the GKR interactive proof scheme, which we now describe.

The Protocol and Round-by-Round Soundness. Let $\{\mathbb{F}_n\}_n$ be a $\mathsf{polylog}(n)$-time explicit representation of finite fields of characteristic two and order $|\mathbb{F}_n| \geq n^{\omega(1)}$, $|\mathbb{F}_n| \leq 2^{\mathsf{polylog}(n)}$. When executed on input $x \in \{0,1\}^n$, the protocol will only involve polynomials over the field $\mathbb{F}_n$, and we omit subscripts accordingly.

Throughout the GKR protocol, both the prover and verifier maintain a list of pending claims. The initial claim, corresponding to the assertion that $x \in \mathcal{L}_{\mathsf{yes}}$, is that $\hat{V}_x^{d(n)}(w_{\mathsf{out}}) = 1$, where w_{out} is the label of the output wire of C_n. In general claims will be of the form $p(\bar{\mathbb{u}}) = \mathbb{v}$ where p is one of the above polynomials, and $\bar{\mathbb{u}}$ and $\mathbb{v}$ are arbitrary.

In each round, the prover and verifier: (1) reduce multiple claims regarding some polynomial p to a *single* claim regarding p, and (2) reduce that claim to several claims about a "simpler" polynomial.

1. Suppose that the currently pending claims are $p(\bar{\mathbb{u}}_1) = \mathbb{v}_1, \ldots, p(\bar{\mathbb{u}}_k) = \mathbb{v}_k$ (k will in fact always be at most 2). For some canonical association of the set $[k]$ with a subset of $\mathbb{F}$, the prover and verifier construct the unique degree $k-1$ polynomial curve for which $\gamma(i) = \bar{\mathbb{u}}_i$ for all $i \in [k]$. The prover sends to the verifier an explicitly represented univariate polynomial g^* that has degree at most $(k-1) \cdot \deg(p)$ and is purportedly equal to $p \circ \gamma$. The verifier checks that $g^*(i) = \mathbb{v}_i$ for each $i \in [k]$, and responds with a random challenge $\mathbb{r} \leftarrow \mathbb{F}$. All claims about p are then replaced with the single claim that $p(\bar{\mathbb{u}}^*) = \mathbb{v}^*$, where $\bar{\mathbb{u}}^* = \gamma(\mathbb{r})$ and $\mathbb{v}^* = g^*(\mathbb{r})$.

2. The polynomial p has a defining equation – either Equation (2.2), (2.3), or (2.4) – that expresses $p(\bar{\mathbb{u}}^*)$ as a function φ applied to a constant number of other polynomial evaluations. The prover sends these other evaluations, and the verifier checks that applying φ yields $\mathbb{v}^*$.

After $r(n) = O\Big(d(n) \cdot s(n)\Big)$ rounds a single claim remains, regarding $\hat{X}_x^{(0)}$. Such a claim is directly checkable by the verifier in $\tilde{O}(n)$ field operations.

Theorem 2.43. *For every promise problem $\mathcal{L} = (\mathcal{L}_{\mathsf{yes}}, \mathcal{L}_{\mathsf{no}})$ in log-space uniform* NC*, there is a public-coin interactive proof Π for $\mathcal{L}$ with verifier running time $\tilde{O}(n)$, prover running time $\mathsf{poly}(n)$, and round-by-round soundness error $\mathrm{negl}(n)$. Moreover, the corresponding relation R_Π is sampleable in $\mathsf{poly}(n)$ time.*

Proof. The prover and verifier efficiency claims follow directly from examination of the above protocol.

We define State so that $\mathsf{State}(x, \tau_{i-1})$ is acc if each pending claim after τ_{i-1} is true, and otherwise $\mathsf{State}(x, \tau_{i-1})$ is rej. All the polynomials involved are evaluable in time $\mathsf{poly}(S)$, which by log-space uniformity is $\mathsf{poly}(n)$, so State is too.

We analyze the round-by-round soundness error of steps 1 and 2 of the protocol, described above.

The round-by-round soundness error incurred in step 1 is the fraction of $\mathbb{r}$'s for which $g^*(\mathbb{r}) = p(\gamma(\mathbb{r}))$. The assumption that the currently pending claims are not all true implies that the polynomials g^* and $p \circ \gamma$ are not equal, so the fraction of "bad $\mathbb{r}$'s" is bounded by $\frac{k \cdot \deg(p)}{|\mathbb{F}|}$. With our choice of $\mathbb{F}$, this is negligible in n.

Step 2 incurs no round-by-round soundness error: if $\mathbb{v}^* \neq p(\bar{\mathbb{u}}^*)$, then at least one of the right-hand-side claims must be false.

To write the relation R_Π more explicitly, we first observe by inspection of Eqs. (2.2) to (2.4) that there is a fixed sequence of polynomials $Q_1, \ldots, Q_r$ such that claims in the i^{th} round are about Q_i. R_Π consists of the pairs $\big((x, \tau|\alpha), \mathbb{r}\big)$ for which:

- α and β are in $\mathbb{F}_n$, where n is the length of x.

- τ is of the form $\alpha_1|\beta_1| \cdots |\alpha_i|\beta_i$ for some $0 \leq i < r(n)$.

- Each α_j is of the form $(\mathbb{v}_{j,1}, \mathbb{v}_{j,2}, g_j)$, for some $\mathbb{v}_{j,1}, \mathbb{v}_{j,2} \in \mathbb{F}_n$ and some g_j that is a degree-$\mathsf{polylog}(n)$ univariate polynomial, represented as a list of coefficients in $\mathbb{F}_n$.

- Each β_j lies in $\mathbb{F}_n$.

- Of the claims "$Q_{i+1}(\bar{\mathbb{u}}_{i+1,1}) = \mathbb{v}_{i+1,1}$" and "$Q_{i+1}(\bar{\mathbb{u}}_{i+1,2}) = \mathbb{v}_{i+1,2}$" that are pending after the prover sends α_{i+1}, at least one claim is false, but $g_{i+1}(1) = \mathbb{v}_{i+1,1}$ and $g_{i+1}(2) = \mathbb{v}_{i+1,2}$.

- $g_{i+1}(\mathbb{r}) = Q_{i+1}\Big((1 - \mathbb{r}) \cdot \mathbb{v}_{i+1,1} + \mathbb{r} \cdot \mathbb{v}_{i+1,2}\Big)$.

The algorithm for sampling R_Π works as follows. Given a transcript $\tau_{i-1}|\alpha_i$, compute the list of pending claims $p(\bar{\mathbb{u}}_1) = \mathbb{v}_1, \ldots, p(\bar{\mathbb{u}}_k) = \mathbb{v}_k$ that follow τ_{i-1}. If all pending claims are correct, then there is nothing to do. Otherwise, let γ denote the unique degree $k-1$ polynomial curve for which $\gamma(i) = \bar{\mathbb{u}}_i$ for all $i \in [k]$, and parse α_i as a univariate polynomial g^*. A verifier message $\beta_i \in \mathbb{F}$ is bad – that is, $\big((x, \tau_{i-1}|\alpha_i), \beta_i\big) \in R_\Pi$ – if and only if $g^*(\beta_i) = p(\gamma(\beta_i))$.

Thus, to sample from R_Π, we just need to output a random root of $g^* - p \circ \gamma$. Using the Cantor-Zassenhaus algorithm [CZ81], we can enumerate all roots with probability $\frac{2}{3}$, and therefore with any probability arbitrarily exponentially close to 1 (i.e., $1 - e^{-\mathsf{poly}(n)}$ for any desired poly). If this factorization succeeds, we can sample an element from the set of all roots with arbitrarily exponentially small sampling error, giving the stated result. □

2.6.3 Publicly Verifiable Delegation for Log-Space Uniform NC

Theorem 2.44. *If Assumption 2.2 holds, then every promise problem in log-space uniform* NC *has a publicly verifiable non-interactive argument scheme with adaptive soundness such that for inputs of length n:*

- *The scheme uses a common random string of length $\tilde{O}(n)$.*
- *Proofs are of length* $\mathsf{polylog}(n)$ *and are generatable in time* $\mathsf{poly}(n)$.
- *Proofs are publicly verifiable in time $\tilde{O}(n)$.*

Proof. Our construction uses the following building blocks.

- The round-by-round sound interactive proof of Theorem 2.43.
- A secret-key encryption scheme $\mathsf{SKE} = (\mathsf{SKE.Gen}, \mathsf{SKE.Enc}, \mathsf{SKE.Dec})$ with keys of length $\kappa = \kappa(\lambda) \geq \lambda^{\Omega(1)}$ and universal ciphertexts that are $2^{-\kappa} \cdot \mathsf{poly}(\kappa)$-

KDM-secure for arbitrary $\mathsf{poly}(\lambda)$-size computable functions of the secret key.[31] Specifically, Assumption 2.2 implies that secret-key Regev encryption satisfies these properties, with secret distribution χ_{sec} that is uniform on $[-B, B)$ for some B specified below and error distribution χ_{err} that is uniform on $[-\frac{q}{4}, \frac{q}{4})$.

Furthermore, the *proof of security* of our delegation scheme uses an additional building block:

- A (secret-key) fully homomorphic encryption scheme FHE that is $2^{-|\mathsf{sk}|} \cdot \mathsf{poly}(|\mathsf{sk}|)$-circular secure. We instantiate FHE using the [BV14] FHE scheme in which the underlying LWE secret and error distributions (χ_{sec} and χ_{err}) are uniform in the range $[-B, B)$ for $B \approx \frac{q}{n^{.51} \max_i |\beta_i|}$. Here, $|\beta_i| = \mathsf{polylog}(n)$ denotes the length of the ith verifier message in the [GKR08] protocol.

Combining Theorem 2.13, Theorem 2.38, Theorem 2.43, and Theorem 2.20, we conclude that the following protocol is a succinct non-interactive argument system for log-space uniform NC.

- **Input**: An instance $x \in \mathcal{L}_{\mathsf{yes}} \cup \mathcal{L}_{\mathsf{no}}$.

- **Common Random String**: A uniformly random string h that describes a Regev ciphertext $\mathbf{ct} \in \mathbb{Z}_q^{(n'+1)\times m}$ where $n' \cdot \lfloor \log(2B + 1) \rfloor$ is at least the length of a [GKR08] transcript (including the input x), and m is at least as large as any verifier message.

- **Proof**: messages α_i computed according to the [GKR08] prover algorithm, where the verifier messages $\beta_1, \ldots, \beta_r$ are computed inductively by first padding the transcript prefix $\tau_j \overset{\mathsf{def}}{=} \alpha_1 | \beta_1 | \cdots | \alpha_j$ so that it can be viewed as an element of $[-B, B)^{n'}$, and then computing $\beta_j = \mathsf{SKE.Dec}(\tau_j, h)$,

- **Verification**: The verifier accepts the transcript $(h, \alpha_1, \ldots, \alpha_r)$ as a proof for x if the GKR verifier algorithm accepts the transcript $\alpha_1 | \beta_1 | \cdots | \beta_{r-1} | \alpha_r$ on input x, where each β_i is computed as above.

[31] If given a time bound T in advance for the computations to be supported in the delegation protocol, there is an explicit polynomial $p(|\mathsf{sk}|)$ that can replace the "arbitrary $\mathsf{poly}(|\mathsf{sk}|)$" condition. However, the description size of the hash function must depend only logarithmically on T.

Security follows from the exponential KDM-security of SKE (and the universal ciphertexts property of SKE, which holds unconditionally), which in turn follows from the exponential circular security of FHE. □

We are able to achieve an even shorter CRS (any n^ϵ rather than $\tilde{O}(n)$) if we are willing to settle for non-adaptive soundness.

Theorem 2.45. *If Assumption 2.2 holds, then for every promise problem $\mathcal{L} = (\mathcal{L}_{\mathsf{yes}}, \mathcal{L}_{\mathsf{no}})$ in log-space uniform* NC *and every $\epsilon > 0$, there is a publicly verifiable non-interactive argument scheme with* non-adaptive *soundness such that for inputs of length n:*

- *The scheme uses a common random string of length $O(n^\epsilon)$.*[32]
- *Proofs are of length* $\mathsf{polylog}(n)$ *and are generatable in time* $\mathsf{poly}(n)$.
- *Proofs are publicly verifiable in time $\tilde{O}(n)$.*

2.7 Non-Interactive Zero Knowledge

We present the construction of Non-Interactive Zero Knowledge (NIZK) Arguments assuming that LWE holds with exponentially small inversion probability (and suitable parameters). We begin by recalling the definition of NIZK.

2.7.1 Non-Interactive Zero Knowledge Arguments

Definition 2.46. *A* non-interactive zero knowledge (NIZK) argument system Π *for an* **NP** *relation R consists of three ppt algorithms* (Setup, P, V) *with the following syntax.*

- $\mathsf{Setup}(1^n)$ *takes as input a statement length n and outputs a common reference string* crs.
- $P(\mathsf{crs}, x, w)$ *takes as input the common reference string, as well as x and w such that $(x, w) \in R$. It outputs a proof π.*

[32] Under stronger but still plausible assumptions, the common random string can instead have length $\mathsf{polylog}(n)$; this would correspond to assuming FHE satisfying almost-optimal security against subexponential-time adversaries.

- $V(\mathsf{crs}, x, \pi)$ *takes as input the common reference string, a statement* x*, and a proof* π*. It outputs a bit* b*. If* $b = 1$*, we say that* V accepts*, and otherwise we say that* V rejects*.*

The proof system Π *must satisfy the following requirements. Recall that* $\mathcal{L}(R)$ *denotes the language* $\{x : \exists w \text{ s.t. } (x, w) \in R\}$ *and* R_n *denotes the set* $R \cap (\{0,1\}^n \times \{0,1\}^*)$*.*

- **Completeness.** *For every* $(x, w) \in R$*, it holds with probability* 1 *that* $V(\mathsf{crs}, x, \pi) = 1$ *in the probability space defined by sampling* $\mathsf{crs} \leftarrow \mathsf{Setup}(1^{|x|})$ *and* $\pi \leftarrow P(\mathsf{crs}, x, w)$*.*

- **Soundness.** *For every* $\Big\{x_n \in \{0,1\}^n \setminus \mathcal{L}(R)\Big\}$ *and every polynomial size* $P^* = \{P^*_n\}$*, there is a negligible function* ν *such that*

$$\Pr_{\substack{\mathsf{crs} \leftarrow \mathsf{Setup}(1^n) \\ \pi \leftarrow P^*_n(\mathsf{crs})}} \Big[V(\mathsf{crs}, x_n, \pi) = 1\Big] \leq \nu(n).$$

- **Zero Knowledge.** *There is a ppt simulator* Sim *such that for every ensemble* $\Big\{(x_n, w_n) \in R_n\Big\}$*, the distribution ensembles*

$$\Big\{\Big(\mathsf{crs}_n, P(\mathsf{crs}_n, x_n, w_n)\Big)\Big\}_n$$

  and

$$\Big\{\mathsf{Sim}(x_n))\Big\}_n$$

  are computationally indistinguishable in the probability space defined by sampling $\mathsf{crs}_n \leftarrow \mathsf{Setup}(1^n)$ *(and evaluating* P *and* Sim *with independent and uniformly randomness).*

  If the distributions are statistically indistinguishable*, then* Π *is said to be* statistically zero knowledge*.*

A NIZK argument system can also satisfy various stronger properties. We list two important variants below.

- **Public Coin (or "Common Random String")**: Π is called $\textsf{public coin}$ (aka, a $\textsf{NIZK in the common } \textsf{\textit{random}} \textsf{ string model}$) if $\textsf{Setup}(1^n)$ simply samples and outputs a uniformly random string.

- **Adaptive Soundness**: Π is $\textsf{adaptively sound}$ if for every polynomial size algorithm $P^* = \{P_n^*\}$, there is a negligible function ν such that for all n,

$$\Pr_{\substack{\textsf{crs} \leftarrow \textsf{Setup}(1^n) \\ (x,\pi) := P_n^*(\textsf{crs})}} [x \notin \mathcal{L}(R) \wedge V(\textsf{crs}, x, \pi) = 1] \leq \nu(n).$$

2.7.2 NIZK from Bounded Correlation Intractability

In this section, we construct NIZK arguments in the common random string (CRS) model from hash families that are correlation intractable with respect to efficiently sampleable relations. We obtain these NIZK arguments by applying the Fiat-Shamir transform to an instantiation of the [GMW86] proof system (repeated in parallel) in which the underlying commitment scheme is encryption under a public key that is included as part of the CRS).

With a generic public-key encryption scheme or with a secret-coin hash family, this approach yields NIZKs with a common *reference* string.[33] However, if the public key encryption scheme and the hash family both have pseudorandom (public) keys, then this approach yields NIZK arguments in the common *random* string model. Also, we show that if encryption under a uniformly random public key[34] is *lossy* [KN08, PVW08, BHY09], then this argument system is *statistical* zero knowledge (rather than just computational zero knowledge). Finally, we note that we can also obtain *adaptive soundness*[35] if the Fiat-Shamir hash function is applied to the concatenation $x||\mathbf{a}$ (where $\mathbf{a}$ is the first message of a three-round protocol) rather than just to $\mathbf{a}$.

We begin by recalling the folklore notion of a "commit-challenge-response" proof

[33] In the common reference string model, the prover and verifier have shared access to a CRS sampled by some trusted setup algorithm. In the common random string model, the CRS is required to be a uniformly random string.

[34] By "uniformly random public key", we mean a public key that is a uniformly random string, rather than a public key sampled according to the key generation algorithm.

[35] if the CRS distribution uses well-formed (rather than lossy) public keys

system. In particular, the [GMW86] protocol for the (NP-complete) problem of 3-coloring falls into this framework. We include for completeness an explicit definition that is taken verbatim from [HL18].

Definition 2.47 (Commit-Challenge-Response Proof System)**.** *A 3-message proof system* $\Pi = (P, V)$ *for a language* L *with witness relation* R *is called* $\mathsf{commit\text{-}challenge\text{-}response}$ *if it satisfies the following properties.*

1. *The first message is sent by the prover to the verifier. This message, which we denote by* $\mathbf{a}$*, consists of a block-wise commitment (under a statistically binding commitment scheme) to a string* y *that is a function of both the common input* x *and the prover's private input* w*.*

2. *The second message, which we denote by* $\mathbf{e}$ *and refer to as the verifier's "challenge", is sent by the verifier to the prover and is sampled uniformly at random from a* $\mathsf{poly}(|x|)$*-size alphabet* Σ*.*

3. *The third and final message, which we denote by* $\mathbf{z}$*, is sent by the prover to the verifier, and consists of a* decommitment *to* y_T*, i.e., a subset* T *of the blocks of* y*. Here,* T *is a function of the challenge* e*.*

4. *The verifier* V *accepts if and only if (1)* $\mathbf{z}$ *is a valid decommitment of* $\mathbf{a}_T$*, and (2) the tuple* $(x, y_T, \mathbf{e})$ *passes some efficient test* Check*, where* y_T *is the value to which* $\mathbf{a}_T$ *was decommitted.*

In order to obtain our result on statistical zero knowledge, we also a define a specific kind of honest-verifier zero knowledge for commit-challenge-response protocols.

Definition 2.48 (Special Honest-Verifier Zero Knowledge)**.** *We say that a commit-challenge-response proof system* Π *is* special honest-verifier zero knowledge *if there is a ppt simulator* SHVSim *that on input* x *produces a string* $(e, y_{T(e)})$ *that is* identical *to the distribution of* $(e, y_{T(e)})$ *where* e *is uniformly random and* y *is produced by the honest proving algorithm* $P(x, w)$*.*

We note that if a commit-challenge-response protocol Π is special honest-verifier zero knowledge, then it is also honest-verifier zero knowledge; the simulator simply runs $\mathsf{SHVSim}(x)$ and then commits to a string $\tilde{y}$ that matches $y_{T(e)}$ in the locations corresponding to $T(e)$ and satisfies $\tilde{y}_j = 0$ otherwise.

Given any commit-challenge-response proof system $\Pi = (P, V)$ and any public key encryption scheme $\mathsf{PKE} = (\mathsf{PKE.Gen}, \mathsf{PKE.PKE.Enc}, \mathsf{PKE.Dec})$, we instantiate the commitment scheme in Π using PKE. That is, Π is augmented with a common reference string pk (a public key sampled using $\mathsf{PKE.Gen}$) and a commitment $\mathsf{com}(\mathsf{pk}, b)$ is sampled by calling $\mathsf{PKE.Enc}(\mathsf{pk}, b)$. The encryption randomness used in the call to $\mathsf{PKE.Enc}(\mathsf{pk}, b)$ serves as a decommitment for the bit b.

We will apply the Fiat-Shamir transform to Π repeated $\lambda \cdot |\Sigma|$ times in parallel.[36] The repeated protocol $\Pi^{\lambda \cdot |\Sigma|}$ consists of three messages $(\mathbf{a}, \mathbf{e}, \mathbf{z})$, and for a fixed secret key sk and instance $x \notin L$, we consider the relation

$$R_{x,\mathsf{sk}} = \left\{ (\mathbf{a}, \mathbf{e}) : \text{Check}\left(x, y^{(i)}_{T(e^{(i)})}, e^{(i)}\right) = 1 \text{ for all } i, \text{ where } \mathbf{y} = \mathsf{Dec}(\mathsf{sk}, \mathbf{a}) \right\}.$$

In [HL18], it was shown that

Imported Theorem 2.49 ([HL18], see Theorem 6.6)**.** *If $\mathcal{H}$ is correlation intractable with respect to all relations of the form $R_{x,\mathsf{sk}}$, then applying the Fiat-Shamir transform to $\Pi^{\lambda \cdot |\Sigma|}$ yields a sound two-message protocol.*

In order to obtain adaptive soundness, we define a new relation R_{sk} as follows:

$$R_{\mathsf{sk}} = \left\{ ((x, \mathbf{a}), \mathbf{e}) : x \notin L \text{ and Check}\left(x, y^{(i)}_{T(e^{(i)})}, e^{(i)}\right) = 1 \text{ for all } i, \text{ where } \mathbf{y} = \mathsf{Dec}(\mathsf{sk}, \mathbf{a}) \right\}.$$

As written, the length of the "output" $\mathbf{e}$ may depend on the input x (i.e. not just its length); however, we can extend this relation by padding the output up to the maximum length of $\mathbf{e}$ as a function of n.

[36] Parallel repetition is done so that the soundness error is reduced to $2^{-\Omega(\lambda)}$.

We first note the following.

Lemma 2.50. R_{sk} *is sparse and non-uniformly efficiently sampleable for every* $(\mathsf{pk}, \mathsf{sk})$ *in the support of* $\mathsf{PKE.Gen}$.[37]

Proof. The sparsity of R_{sk} follows from the $2^{-\Omega(\lambda)}$-soundness of $\Pi^{\lambda \cdot |\Sigma|}$.[38] To see this, note that because Π is sound, we have in particular that for every $x \notin L$ and every string y, with $1 - 2^{-\Omega(\lambda)}$ probability over the choice of $\mathbf{e}$, we have that $\mathrm{Check}\left(x, y^{(i)}_{T(e^{(i)})}, e^{(i)}\right) = 0$ for some i. Therefore, we have that for every $\mathbf{a}$, the same statement holds for $y = \mathsf{PKE.Dec}(\mathsf{sk}, \mathbf{a})$ (and every $x \notin L$). Thus, R_{sk} is $2^{-\Omega(\lambda)}$-sparse.

To see that R_{sk} is efficiently sampleable, we note that given $x, \mathbf{a}$ and sk, we can compute $y = \mathsf{Dec}(\mathsf{sk}, \mathbf{a})$; then, for each block $y^{(i)}$, we can enumerate over all challenges $e^{(i)}$, compute $\mathrm{Check}(x, y^{(i)}_{T(e^{(i)})})$, and then sample a uniformly random $e^{(i)}$ subject to passing the check. □

We will use this fact to construct a NIZK argument system for $\mathbf{NP}$ assuming public-key encryption and *programmable* hash functions that are correlation intractable for all efficiently sampleable relations. This follows the NIZK constructions of [CCRR18, HL18]. In addition, and as noted above, we prove that for special PKE schemes such as Regev encryption, the NIZK can be made to satisfy *statistical* zero knowledge and rely on a common *random* string.

Construction 2.51. *Suppose that:*

- $\Pi = (P, V)$ *is a commit-challenge-response proof system for a language* L*,*
- $\mathsf{PKE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *is a public key encryption scheme, and*
- $\mathcal{H}$ *is a hash family.*

We then define $\mathsf{NIZK}^{\Pi, \mathsf{PKE}, \mathcal{H}}_{\mathrm{FS}} = (\mathsf{Setup}, \tilde{P}, \tilde{V})$ *as follows.*

[37] It is worth noting that R_{sk} may not be efficiently *decidable*, as this would require deciding whether $x \in L$. We only need to be able to sample a uniformly random "bad" challenge when promised that $x \notin L$.

[38] We technically need the fact that $2^{-\Omega(\lambda)}$-soundness holds for every fixed choice of $(\mathsf{pk}, \mathsf{sk})$.

- Setup*: On input* 1^n*, the setup algorithm samples* $\mathsf{pk} \leftarrow \mathsf{Gen}(1^n)$ *and* $h \leftarrow \mathcal{H}_n$*, and then outputs the common reference string* (pk, h)*.*

- $\tilde{P}$*: On input* $\big((\mathsf{pk}, h), x, w\big)$*, the prover* $\tilde{P}$ *generates a proof* π *that consists of:*

  - $\lambda \cdot |\Sigma|$ *independently sampled first messages (commitments)* $\mathbf{a} = \big(a^{(1)}, \dots, a^{(\lambda \cdot |\Sigma|)}\big)$ *that arise from instantiating* P *with the non-interactive commitment* $\mathsf{Enc}(\mathsf{pk}, \cdot)$*.*

  - *The responses* $\mathbf{z} = \big(z^{(1)}, \dots, z^{(\lambda \cdot |\Sigma|)}\big)$ *of* P *that correspond to the* $\lambda \cdot |\Sigma|$ *challenges* $\mathbf{e} = (\mathbf{e}^{(1)}, \dots, \mathbf{e}^{(\lambda \cdot |\Sigma|)})$ *obtained as an appropriate-length prefix of* $h(x||\mathbf{a})$*.*

- $\tilde{V}$*: On input* $\big((\mathsf{pk}, h), x, \pi\big)$*, the verifier accepts iff* V *accepts the* $\lambda \cdot |\Sigma|$ *transcripts* $\big(\mathsf{pk}, x, a^{(i)}, e^{(i)}, z^{(i)}\big)$ *where* $\mathbf{e}$ *is again the first* $\lambda \cdot |\Sigma| \cdot \log(|\Sigma|)$ *bits of* $h(x||\mathbf{a})$*.*

Theorem 2.52. *Suppose that:*

- $\Pi = (P, V)$ *is an honest-verifier zero knowledge commit-challenge-response proof system for an* **NP** *language* L*.*

- $\mathsf{PKE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *is a public key encryption scheme.*

- $\mathcal{H}$ *is a hash family (with appropriate input and output lengths) that is correlation intractable for all efficiently sampleable relations, and in addition satisfies the following additional property:*

  - **Approximate Average-Case Programmability***: There is an efficient sampling algorithm* $h \leftarrow \mathsf{Samp}(\mathbf{a}, \mathbf{e})$ *such that for any fixed* $\mathbf{a}$*, the distribution* $\{h \leftarrow \mathsf{Samp}(\mathbf{a}, \mathbf{e})\}$ *for uniformly random* $\mathbf{e}$ *is statistically indistinguishable from* $h \leftarrow \mathcal{H}$*.*

Then, the protocol $\widetilde{\Pi}$ *(as in Construction 2.51) is an adaptively sound* NIZK *argument scheme for* L*.*

Moreover:

1. *If public keys* pk *generated using* $\mathsf{PKE.Gen}$ *are (computationally) pseudorandom and* $\mathcal{H}$ *has (computationally) pseudorandom keys, then* $\widetilde{\Pi}$ *is a (non-adaptively sound)* NIZK *when the CRS is instead sampled to be a uniformly random string.*

2. *If a uniformly random public key* pk *of the scheme is* lossy *– meaning that* $(\mathsf{pk}, \mathsf{Enc}(\mathsf{pk}, 0)) \approx_s (\mathsf{pk}, \mathsf{Enc}(\mathsf{pk}, 1))$ *when* pk *is sampled uniformly at random – and* Π *satisfies special honest-verifier zero knowledge, then* $\widetilde{\Pi}$ *is a (non-adaptively sound) non-interactive* statistical *zero knowledge (NISZK) argument system.*

3. *If condition (2) holds and* $\mathcal{H}$ *has* statistically *pseudorandom keys, then* $\widetilde{\Pi}$ *is a (non-adaptively sound) NISZK argument when the CRS is sampled to be a uniformly random string.*

Proof. **Completeness** of the protocol follows directly from the completeness of Π.

We next argue **(adaptive) soundness**. Suppose that some efficient algorithm $\mathcal{A}$, given (pk, h), is able to produce $(x, \mathbf{a}, \mathbf{z})$ such that, with non-negligible probability, it holds that $x \notin L$ and $\widetilde{\Pi}.V(\mathsf{pk}, h, x, \mathbf{a}, \mathbf{z}) = 1$. We then define the following algorithm $\mathcal{A}'$ breaking the correlation intractability of $\mathcal{H}$.

- $\mathcal{A}'$ first samples $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{PKE.Gen}(1^\lambda)$ and chooses the relation R_{sk} defined as above.

- $\mathcal{A}'$ is then given a hash function $h \leftarrow \mathcal{H}$. It runs $\mathcal{A}(\mathsf{pk}, h)$, obtaining $(x, \mathbf{a}, \mathbf{z})$ and outputs $(x, \mathbf{a})$.

To see that this breaks the correlation intractability of $\mathcal{H}$ with respect to R_{sk}, we note that whenever $\widetilde{\Pi}.V(\mathsf{pk}, h, x, \mathbf{a}, \mathbf{z}) = 1$, $\mathbf{z}$ must contain valid decommitments to some strings $\tilde{y}^{(i)}_{T(e_i)}$ for each i (where $\mathbf{e}$ is computed as in Construction 2.51), which are necessarily the corresponding blocks of $\mathsf{PKE.Dec}(\mathsf{sk}, \mathbf{a})$ by perfect decryption correctness. Then, the fact that $\widetilde{\Pi}.V(\mathsf{pk}, h, x, \mathbf{a}, \mathbf{z}) = 1$ implies by definition that $R_{\mathsf{sk}}(x, \mathbf{a}, h(x, \mathbf{a})) = 1$.

Therefore, since we know by Lemma 2.50 that R_{sk} is sparse and efficiently sampleable and $\mathcal{H}$ is correlation intractable for all such relations, we conclude that $\widetilde{\Pi}$ is adaptively sound.

If the CRS is instead sampled to be a uniformly random string and PKE and $\mathcal{H}$ have pseudorandom (public) keys, then non-adaptive soundness follows by a hybrid argument: if an efficient cheating prover could break the (non-adaptive) soundness of the protocol with a uniformly random CRS, then the same prover would break (non-adaptive) soundness of the protocol $\widetilde{\Pi}$ where the CRS is generated using $\mathsf{PKE.Gen}$ and $\mathcal{H}.\mathsf{Gen}$. This would contradict soundness of the basic protocol, hence the modified protocol is sound. Note that this argument only shows that the modified protocol is non-adaptively sound, because the win condition of the adaptive soundness game is not efficiently checkable.

Finally, we show that our scheme is **zero knowledge**. To do so, we write down the following simulator $\mathsf{Sim}(x, \mathsf{pk})$:

- Given x, first sample a uniformly random challenge vector $\mathbf{e}$.
- Then, run the honest verifier simulator $\Pi.\mathsf{HVSim}(x, \mathsf{pk}, \mathbf{e})$ associated to Π to produce a simulated first message $\mathbf{a}$ and third message $\mathbf{z}$
- Finally, sample a hash function h using the sampler $\mathsf{Samp}(\mathbf{a}, \mathbf{e})$ and output $(\mathsf{CRS}, \mathbf{a}, \mathbf{z})$ where $\mathsf{CRS} = (\mathsf{pk}, h)$.

The claim is that when $x \in L$ and pk is generated using $\mathsf{PKE.Gen}$, $\mathsf{Sim}(x, \mathsf{pk})$ is computationally indistinguishable from an honest proof (using x and a witness w). This follows by a hybrid argument. First, we note that $(\mathbf{a}, \mathbf{e}, \mathbf{z})$ as sampled by HVSim is computationally indistinguishable from an honest proof $(\mathbf{a}, \mathbf{e}, \mathbf{z})$ (using a uniformly random $\mathbf{e}$) by the simulation security of Π, which implies that the output of $\mathsf{Sim}(x, \mathsf{pk})$ is computationally indistinguishable from $(\mathsf{CRS}, \mathbf{a}, \mathbf{z})$ where $(\mathbf{a}, \mathbf{e}, \mathbf{z})$ is an honest proof and h is sampled from the distribution $h \leftarrow \mathsf{Samp}(\mathbf{a}, \mathbf{e})$. The approximate average-case sampleability of $\mathcal{H}$ then implies that this distribution is indistinguishable from an honest (CRS, proof) pair in the round-compressed protocol.

Finally, suppose that PKE is a lossy encryption scheme in which lossy public keys are uniformly random. We again consider the modified protocol in which the public key portion of the CRS is sampled uniformly at random, and our simulator will operate as follows.

- Sample a public key pk uniformly at random.
- Repeatedly call the special simulator $\Pi.\mathsf{SHVSim}(x)$, producing $(e^{(i)}, y^{(i)}_{T(e^{(i)})})_i$.
- Set $\mathbf{a}$ to be a commitment to strings $\tilde{y}^{(i)}$ matching the substrings above (and 0 otherwise), and $\mathbf{z}$ to be decommitments to $(y^{(i)}_{T(e^{(i)})})_i$.
- Sample $h \leftarrow \mathsf{Samp}(\mathbf{a}, \mathbf{e})$.

In this situation, the commitment scheme used to instantiate Π is actually *statistically hiding* by the lossiness of PKE, which implies that the simulated distribution $(\mathbf{a}, \mathbf{e}, \mathbf{z})$ is statistically indistinguishable from a honest (parallel repeated) Π-proof. This implies that our simulated proof $(\mathsf{pk}, h, \mathbf{a}, \mathbf{z})$ is statistically indistinguishable from the distribution $(\mathsf{pk}, h, \mathbf{a}, \mathbf{z})$ in which $(\mathbf{a}, \mathbf{e}, \mathbf{z})$ is an honest (parallel repeated) Π-proof and h is sampled from $h \leftarrow \mathsf{Samp}(\mathbf{a}, \mathbf{e})$. Then, the approximate average-case sampleability of $\mathcal{H}$ (along with the fact that $\mathcal{H}$ has statistically pseudorandom keys) again tells us that this is statistically indistinguishable from an honest proof in the round-compressed protocol. This completes the proof of statistical zero knowledge, and of Theorem 2.52. □

Instantiations

If the (standard) LWE assumption holds, then a variant of Regev public-key encryption satisfies all the conditions required by Theorem 2.52 to ensure that the resulting NIZK argument is statistically zero knowledge in the common *random* string model:

- Regev public-key encryption [Reg05] is a lossy public key encryption scheme.
- To ensure that decryption is perfectly correct, we will use a truncated Gaussian distribution for the noise distribution in our variant of Regev encryption. The

polynomial security of this variant (which is all that we require of our commitment scheme) follows from this follows from the security of standard Regev encryption, i.e. from LWE.

The hash family $\mathcal{H}$ in Theorem 2.52 can be instantiated using any of the KDM-secure encryption schemes from Section 2.4.2 or Section 2.4.3. It is clear by inspection that the hash family from Section 2.4.2 satisfies (perfect) programmbility (this was already noted in [CCRR18]). Moreover, the hash families from Section 2.4.3 satisfies approximate programmability. An approximate sampling algorithm for the hash family using a secret-key Regev (or [ACPS09]) encryption scheme $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ and randomized encoding scheme $(\mathsf{RE.Enc}, \mathsf{RE.Dec}, \mathsf{RE.Sim})$ samples $h \leftarrow \mathsf{Samp}(\mathbf{a}, \mathbf{e})$ by calling $\mathbf{E} \leftarrow \mathsf{RE.Sim}(\mathbf{e})$ and then sampling from the conditional distribution $h \mid \mathsf{Dec}(\mathbf{a}, h) = \mathbf{E}$. If the randomized encoding is $(1 - \text{negl}(\lambda))$-approximately blind, then this sampling algorithm satisfies the desired property.

2.7.3 Our NIZK Protocol

We conclude this section by giving an explicit description of our NIZK protocol.

Theorem 2.53. *If Assumption 2.1 holds with modulus $q = pq'$ for some prime p, secret distribution uniform over $[-\frac{p}{2}, \frac{p}{2})$, and noise distribution uniform over $[-\frac{q'}{2}, \frac{q'}{2})$, then every language $\mathcal{L} \in \mathbf{NP}$ has a (publicly verifiable)* NIZK *argument scheme Π. Moreover, Π can be chosen to have either adaptive soundness or statistical zero knowledge.*

Proof. For simplicity, we describe the NIZK argument system assuming the exponential hardness of **Search-LWE** with binary secrets, but our argument system that considers **Search-LWE** for larger secrets follows the same blueprint.

Our NIZK argument scheme for **NP** uses the following building blocks.

- The 3-coloring protocol of [GMW86].
- A public-key encryption scheme $\mathsf{PKE} = (\mathsf{PKE.Gen}, \mathsf{PKE.Enc}, \mathsf{PKE.Dec})$ with perfect decryption correctness, which we instantiate with standard public-key

Regev encryption. The only constraint placed on this instantiation is that the error distribution χ_{err} for this scheme must be $\frac{q}{4n}$-bounded with probability 1.

- A secret-key encryption scheme $\widetilde{\mathsf{SKE}} = (\widetilde{\mathsf{SKE}}.\mathsf{Gen}, \widetilde{\mathsf{SKE}}.\mathsf{Enc}, \widetilde{\mathsf{SKE}}.\mathsf{Dec})$ with universal ciphertexts that is $2^{-|\mathsf{sk}|}\mathsf{poly}(|\mathsf{sk}|)$-KDM-secure for $S(|\mathsf{sk}|)$-size computable functions of the secret key, where $S(|\mathsf{sk}|)$ is an explicit polynomial function dictated by the protocol below.

In order to instantiate $\widetilde{\mathsf{SKE}}$, we use two additional building blocks

- A secret-key encryption scheme $\mathsf{SKE} = (\mathsf{SKE.Gen}, \mathsf{SKE.Enc}, \mathsf{SKE.Dec})$ with universal ciphertexts that is $2^{-|\mathsf{sk}|} \cdot \mathsf{poly}(|\mathsf{sk}|)$-KDM secure for key-dependent messages that are $\mathbb{Z}_2$-linear functions of the secret key. This is instaniated with secret-key Regev encryption in which the secret $s \leftarrow \{0,1\}^n$ is a uniformly random binary string, and the error distribution χ_{err} is uniform on the set $[-\frac{q}{4}, \frac{q}{4})$ (and q is even).
- A randomized encoding scheme $\mathsf{RE} = (\mathsf{RE.Enc}, \mathsf{RE.Dec}, \mathsf{RE.Sim})$ for $\mathsf{P/poly}$ that is perfectly blind and $2^{-\omega(n\log(q))}$-secure. This is instantiated with point-and-permute garbled circuits (see Imported Theorem 2.26) instantiated with a subexponentially-secure one-way function.[39]

Combining Theorem 2.13, Theorem 2.52, Theorem 2.25, and Imported Theorem 2.26, we conclude that the following protocol is a NIZK argument scheme for **NP**. In fact, it relies on a common *random* string and satisfies *statistical* zero knowledge.

- **Input:** A graph $x = (V, E)$. The prover receives as additional input a 3-coloring w of x.
- **Common Random String:** A pair (pk, h), where $|\mathsf{pk}|$ is the length of a Regev public key and $|h|$ is the length of a $\widetilde{\mathsf{SKE}}$ ciphertext corresponding to a message of length $\lambda \cdot |\mathsf{RE.Sim}(0^{O(\log(|x|))})|$.

[39] In particular, the existence of such a function (family) follows trivially from the exponential LWE-hardness assumed for the security of SKE.

- **Proof:** A proof π consists of
  - A sequence of $\lambda \cdot |\Sigma|$ independently sampled first messages $\mathbf{a} = (a^{(1)}, \ldots, a^{(\lambda|E|)})$ using the [GMW86] proof system, where commitment is instantiated using $\mathsf{PKE.Enc}$.
  - Responses $\mathbf{z} = (z^{(1)}, \ldots, z^{(\lambda|E|)})$ using the [GMW86] proof system when provided $\lambda|E|$ challenges $\mathbf{e}$ consisting of the first $\lambda \cdot |E| \cdot \log(|E|)$ bits of $\mathsf{RE.Dec}(\mathsf{SKE.Dec}(x||\mathbf{a}, h))$.
- **Verification:** The verifier accepts π if the [GMW86] verifier accepts the $\lambda \cdot |E|$ transcripts $(\mathsf{pk}, x, a^{(i)}, e^{(i)}, z^{(i)})$ where $\mathbf{e}$ computed as above. □

2.8 Success probability of polynomial time algorithms on LWE

We provide a survey of the existing algorithms for breaking LWE and their success probabilities when restricted to run in polynomial time. Recall from Assumption 4.11, we assume the success probability of a polynomial time secret-recovery attack is at most $|\mathrm{Supp}(\chi_{\mathrm{sec}})|^{-n} \cdot \mathsf{poly}(n, \log(q))$. For example, achieving the success probability of $2^{-0.99\lambda}$ would violate this assumption (w.r.t. a search space of size 2^{λ}).

Loosely speaking, all known algorithms for LWE use one or more of the following techniques:

- Lattice basis reduction (e.g. [LLL82, Sch87, SE94]),
- Enumeration (since [Kan87])
- Sieving (since [AKS01])
- Combinatorial (since [BKW03])
- Algebraic (since [AG11]).

These algorithms are typically optimized to run in the smallest possible running time while still solving LWE with overwhelming (or at least noticeable) probability. In contrast, we are concerned with the complexity of solving LWE with tiny (but non-trivial) probability. It is in general not clear if existing algorithms can be adapted to this setting. In particular, we do not know of any way to scale enumeration, sieving, or combinatorial algorithms down to the polynomial-time regime while achieving better success probability than guessing. Let us remark that any polynomial time algorithm with success probability of $2^{-c\lambda}$ can be turned into an algorithm that in $\tilde{O}(2^{c\lambda})$ time and *polynomial space* that succeeds with overwhelming probability, which would be a surprising improvement to these types of algorithms.

We further narrow down the scope of our discussion by restricting each entry of the error vector $\mathbf{e}$ to be sampled from a distribution of standard deviation σ greater or equal to $2\sqrt{n}$. This is justified by the worst-case to average-case reduction [Reg05] which requires σ to be greater or equal to $2\sqrt{n}$, and the Arora-Ge attack [AG11] which is only effective when $\sigma < O(\sqrt{n})$. Let us remark that the Arora-Ge attack also requires sufficiently many samples. Meanwhile, [MP13] shows when limited number of LWE samples are given out, LWE with small errors is as secure as standard LWE. Still, we choose to restrict ourselves to the high noise regime, given that we need the search space of the noise to be larger than the one for the secret anyway.

Let us further remark that when choosing a composite modulus q, additional care has to be taken on the secret and error distributions to avoid the attack by guessing a CRT component of each entry. Consider the following example.[40] Let $q = q_1 \cdot q_2$, where q_1 is a prime of polynomial size and $q_2 = q_1 + 1$. Let χ_{sec} be uniformly over $[0, q_1) \cap \mathbb{Z}$, χ_{err} be uniformly over $[0, q_2) \cap \mathbb{Z}$ (the dimension n of the secret vector will then be the nearest integer of $\lambda / \log q_1$). Then to find the secret vector, it suffices to guess the error vector modulo q_1, and the error distribution is biased modulo q_1. So by always guessing n entries of the error vector to be $\mathbf{0}$ modulo q_1, the probability of winning is $(2/q_1)^n > 2^{-\lambda + \lambda / O(\log(\lambda))}$, violating our assumption. Picking q_2 to be sufficiently large, or picking q_2 to be a multiple of q_1, avoids this error-guessing attack.

[40] We thank Oded Regev and Noah Stephens-Davidowitz for pointing out this vulnerability to us.

2.8.1 The success probability of the lattice basis reduction approach

In the rest of the survey we analyze the success probabilities of the basis reduction algorithms. The flexible parameters in the LWE instance are the secret distribution χ_{sec}, the modulus q, and noise/modulus ratio. We assume the secret distribution is uniform over $[-B, B]^n$ where B is a bound that is typically much smaller than $q/2$, and $(2B+1)^n$ is chosen to be close to 2^λ.

Given an n-dimensional lattice $\mathcal{L}$. The quality of the basis $\mathbf{B}$ produced by a lattice basis reduction algorithm is typically measured by the root Hermite factor δ, defined as $\left(\frac{\|\mathbf{b}_1\|}{\det(\Lambda)^{1/n}}\right)^{1/n}$ where $\mathbf{b}_1$ is the shortest vector in $\mathbf{B}$. The probabilistic polynomial time version of the LLL algorithm [LLL82] achieves $\delta = 1.0746$ in the worst case. Furthermore, Schnorr's algorithm offers a trade-off of finding a $2^{n/k}$-approximate shortest vector with the running time 2^k [Sch87]. Within polynomial time, Schnorr's algorithm outputs a $2^{O\left(\frac{n \log\log n}{\log n}\right)}$-approximate shortest vector in $\mathcal{L}$.

In practice, it is widely observed that the basis reduction algorithms perform much better than the worst-case bound in theory. Nguyen and Stehlé [NS06] suggest that the root Hermite factor achieved by LLL is 1.02 on average. So to give a proper estimation of the hardness of LWE, we consider both the theoretical bounds and the experimental evidences.

Choosing a proper basis. Let $\mathbf{A} \in \mathbb{Z}_q^{m\times n}$, $\mathbf{y} = \mathbf{A}\mathbf{s} + \mathbf{e} \pmod q$ be our target LWE instance. Considering the following lattice $\mathcal{L}_{\mathbf{A}}$ with basis $\mathbf{B}$:

$$\mathbf{B} = \begin{pmatrix} q\mathbf{I}^{m\times m} & \mathbf{A} \\ \mathbf{0} & \mathbf{I}^{n\times n} \end{pmatrix}.$$

Expressing $\mathbf{y}$ as $\mathbf{A}\mathbf{s} + \mathbf{e} + q\mathbf{k}$ gives us $\mathbf{B} \cdot \begin{pmatrix} \mathbf{k} \\ \mathbf{s} \end{pmatrix} - \begin{pmatrix} \mathbf{y} \\ \mathbf{0} \end{pmatrix} = \begin{pmatrix} -\mathbf{e} \\ \mathbf{s} \end{pmatrix}$. If $\|\mathbf{s}\|$ is small (which is the interesting case in our applications), then LWE can be solved by running a

CVP solver on given the basis $\mathbf{B}$ and target $\mathbf{t} := \begin{pmatrix} \mathbf{y} \\ \mathbf{0} \end{pmatrix}$, or running an SVP solver on $\begin{pmatrix} \mathbf{B} & \mathbf{t} \\ \mathbf{0} & M \end{pmatrix}$ where M is a relatively small integer (e.g. $M = 1$). This is referred to as the primal approach.

Alternatively, we can try to solve the SIS problem for $\mathbf{A}$, then conduct a distinguishing attack. This is referred to as the dual approach.

For both approaches, when m (i.e. the number of LWE samples) is sufficiently large, the success probability (or the running time) of the basis reduction algorithm can be optimized by throwing away a few samples and working with a smaller m. From now we assume m is the optimized number of LWE samples. According to [MR09], for the dual approach, given a desired root Hermite factor δ, the optimal choice for m is to set $m \approx \sqrt{n \log q / \log \delta}$, then the state-of-art basis reduction algorithm outputs a vector of length $\min\{q, 2^{2\sqrt{n \log q \log \delta}}\}$. For the primal approach the estimation is similar.

The distribution of the reduced basis. Recall our goal is to estimate the success probability of the secret-recovery attack, in an extreme setting where a success probability of say $2^{-0.99\lambda}$ would be considered non-trivial w.r.t. a search space of size 2^{λ}. So we would like to estimate the probability of finding a "significantly short" vector via the basis reduction algorithms. To keep the discussion concrete, we stick with the following meaning of "significantly short": the root Hermite factor is $(1 + \epsilon)$ where $\epsilon > 0$ is an arbitrarily small constant.

However, understanding the distribution of the outputs produced by LLL/BKZ is known as a challenging problem. Below we survey a few recent studies that tackle the problem from different directions. Jumping ahead, currently we are not able to draw a solid conclusion from these studies to our assumption.

Fixing two target root Hermite factors $\delta_0 > \delta_1 > 1$. Suppose the LLL/BKZ algorithm outputs a *random* basis among all the δ_0-reduced bases (under a well-defined probability measure), then the probability of achieving root Hermite factor δ_1

can be estimated by counting the number of δ_1-reduced bases out of all the δ_0-reduced bases. To this end, Kim and Venkatesh [KV16] study the statistical behavior of δ-Siegel-reduced bases (the Siegel-reduced bases satisfy a slightly weaker condition than the LLL-reduced bases). Their study shows that most of the δ-Siegel-reduced bases have root Hermit factors very close to δ. Formally, let $N_\delta(L)$ be the number of the Siegel-reduced bases for a lattice L of n-dimension with reduction parameter δ. The expectation of $N_\delta(L)$ satisfies $\lim_{\mathrm{n}} \frac{\log \mathrm{EN}_\delta(\mathrm{L})}{\mathrm{n}^3} = \frac{1}{6}\log\delta$. Assuming Riemann hypothesis, the standard deviation of $N_\delta(L)$ is at most $e^{-O(n^2)}$ times its mean. This means for a fixed lattice L, by Chebyshev's inequality, with probability greater than $1 - e^{-O(n^2)}$, the portion of $(\delta_0 - 0.0001)$-reduced bases out of all the δ_0-reduced bases is $e^{-O(n^3)}$.

However, the result of [KV16] indeed justifies that the bases produced by the LLL/BKZ algorithm in practice are largely biased, since otherwise the average root Hermite factor would be closer to 1.0746 but not 1.02. The precise statistical behavior of LLL/BKZ remains largely elusive. Recent experimental studies (cf. [GN08, CN11, MW16, YD17], and more) provide more predictions on the standard deviation and other parameters, which suggest that the basis reduction algorithms might produce an extremely short vector "more often than expected". But at this moment, we are not able to conclude that the basis reduction algorithms achieve root Hermite factor $(1+\epsilon)$ for an arbitrarily small constant $\epsilon > 0$ with non-trivial probability.

Summary. Under the current understanding of the statistical behavior of LLL/BKZ, if the modulus q is chosen to be smaller or equal to $2^{\mathsf{polylog}(n)}$, then the existing lattice reduction algorithms do not seem to achieve non-trivial success probabilities in breaking LWE. As a precautionary measure, the modulus q can be chosen as a polynomial in n, which implies the modulus/noise ratio is polynomial. All of the applications in our paper can use such a choice of q.

Chapter 3

Non-Interactive Zero Knowledge and Correlation Intractability from Circular-Secure FHE

3.1 Introduction

Zero-knowledge (ZK) protocols, introduced by [GMR85], have been ubiquitous in cryptography for the last 30 years. At a high level, a zero-knowledge protocol Π is an interactive protocol between a prover P and a verifier V, in which the verifier V is *convinced* that some statement "$x \in L$" is true but learns *nothing* beyond this fact. This "zero knowledge" property is formalized by the simulation paradigm: proving that an interaction between an honest prover P and any (potentially dishonest) verifier V^* can be *simulated* given only the verifier V^* and its input.

An important and extremely useful variant of zero-knowledge protocols is a *non-interactive zero-knowledge* (NIZK) protocol, in which a proof consists of a single message from the prover to the verifier. While it is known [GO94] that such NIZKs (or even 2 message zero-knowledge argument systems) for languages outside BPP do not exist in the plain model, we can construct NIZK proof systems in a setting where the prover and the verifier have access to a *common reference string* which is chosen from a predefined distribution, e.g. [BFM88, FLS90, CHK03, GOS06, SW14, BP15]. In this work, we make progress on two related open problems in the study of non-interactive zero-knowledge protocols.

Lattice-Based Non-Interactive Zero Knowledge. While it is known how to construct NIZKs for **NP** under standard number-theoretic assumptions such as factoring and Bilinear Diffie-Helllman in prime-order elliptic-curve groups [BFM88, FLS90, CHK03], we do not know how to construct NIZK protocols based on *lattice assumptions* [Ajt96, AD97, Reg05] (except for extremely strong assumptions that suffice for indistinguishability obfuscation). In particular, we do not know how to construct NIZK protocols from any known variant of the learning with errors (LWE) problem [Reg05]. This stands in sharp contrast to the large body of work ([GPV08, PVW08, Gen09, BV11, BLMR13, GVW13, GKP$^+$13, BV15, CC17, GKW17a, WZ17, GKW18], to name a few) that successfully constructed a variety of cryptographic applications from LWE and closely related lattice assumptions — including many applications where LWE-based realizations are the only known ones. In fact, NIZK has stood out as possibly *the* exceptional core cryptographic primitive that can be constructed from the above number-theoretic assumptions (which are notably all broken by polynomial-time quantum computers) but not lattice assumptions.

NIZK via the Fiat-Shamir Transform. A natural approach to constructing NIZK protocols is to use the Fiat-Shamir transform [FS87], which prescribes a general way to remove interaction from public-coin interactive proofs: To transform an interactive proof Π to a non-interactive one, have the verifier first send a hash function h to the prover, and then have the prover compute the entire transcript of Π by itself, replacing the verifier's challenges by the result of applying the hash function to the transcript so far (or portions thereof). The prover sends this entire transcript to the verifier in one message, and the verifier accepts if all checks verify.

Fiat and Shamir proposed to apply this methodology to a three-round identification protocol, using a fixed hash function such as $h(x) = \mathrm{DES}_x(0)$, with the goal of obtaining a signature scheme; later instantiations used SHA and other cryptographic hash functions. As heuristic evidence for its security, Bellare and Rogaway [BR93] showed that when applied to a three-round honest-verifier Zero-Knowledge protocol with negligible soundness error, the Fiat-Shamir transform yields a NIZK protocol for the same language, *as long as the hash function is modeled as a random oracle.* Still, while this paradigm seems like a natural and attractive way to construct simple and efficient NIZK protocols, finding explicit hash functions that suffice to make the approach work under well-defined hardness assumptions has proved to be elusive.

Furthermore, several works have demonstrated that such a hash function would have implications elsewhere, and might also be hard to come by. Specifically, [DNRS99] show that if there is a hash function $\mathcal{H}$ instantiating the Fiat-Shamir transform for some three round public coin interactive proof Π, then Π is *not* (general verifier) zero knowledge. [Bar01, GK03] show that there exists some (artificially constructed) 3-round public-coin *computationally sound* proof (a.k.a an argument) Π, for which the Fiat-Shamir heuristic fails to preserve soundness no matter what hash function is used to instantiate it. Furthermore, [BDG+13] show that no hash function family can be shown to suffice for the Fiat-Shamir via black-box reduction to a game-based assumption, even if one restrict attention to the case where the initial protocol is a three-round statistically sound proof. Nevertheless, it remains plausible that the Fiat-Shamir heuristic could be securely instantiated via some explicit hash family for specific classes of protocols. Showing that this is the case under standard assumptions is a long-standing open problem [BLV03].

Correlation Intractability and Recent Progress. Another hardness property for hash functions, which turns out to be easier to formalize and closely related to "soundness for the Fiat-Shamir transform," is *correlation intractability* (which was defined in [CGH98] for a different purpose). Roughly speaking, a hash function family $\mathcal{H}$ is correlation intractable (CI) for a relation $R(x, y)$ if it is computationally hard, given a random hash key k, to find any input x such that $(x, H_k(x)) \in R$. The most general class of relations typically considered is the set of all sparse relations: a relation R is sparse if for every x, the set of all y such that $(x, y) \in R$ is a negligible faction of all possible values y. As observed in [HMR08], CI families (for this broadest possible class of relations) suffice for the soundness of the Fiat-Shamir transform, whenever the initial protocol is a statistically sound proof.

Initially this observation was taken as evidence for the hardness of constructing CI functions. Recently, however, a number of explicit hash function families were shown to be CI for certain classes of relations under well-defined assumptions [CCR16, KRR17, CCRR18, HL18, CCH+18]. Moreover, these hash functions were shown to be sound for the Fiat-Shamir transform for large classes of protocols. While these are significant advancements, the hardness assumptions used in these works are very strong and not well understood. See more discussion in Section 3.1.2.

3.1.1 Our Contributions

Our main result is a correlation-intractable hash family for a large class of relations, based on circular-secure fully homomorphic encryption (FHE). This is the first construction based on a "fully falsifiable" assumption: one defined via a game between an adversary and a polynomial-time challenger where we assume that every polynomial-time adversary has at most a negligible advantage in the game. Moreover, it is a cryptographic assumption that is widely used elsewhere; in particular, it is currently an essential step for obtaining fully (non-leveled) homomorphic encryption in the first place.

Our correlation-intractable hash family is powerful enough to instantiate the Fiat-Shamir transform for a certain class of public-coin proof systems. The class is quite broad; in particular, it suffices for obtaining NIZKs for all of **NP**. We provide two variants of this transformation: one variant results in NIZK protocol where the zero-knowledge property is *statistical,* and the CRS is "public coin" (in fact, it is uniformly distributed). The other variant results in a NIZK with *statistical soundness.* The latter variant is especially surprising since, even in the random oracle model, the Fiat-Shamir transform only provides computational soundness and therefore our hash function has some advantages even over a random oracle. Furthermore, the two variants have reference strings that are indistinguishable from each other, so the resulting NIZK protocol has a "dual mode" property [DN01, GOS06].

In addition to the NIZK application, we show two other interesting applications of our hash family. One result – essentially following from [DNRS99] – is that assuming circular-secure FHE, a class of natural three-message public-coin protocols (which in particular includes the [GMR85] Quadratic Residuosity protocol) are *not* zero knowledge when repeated in parallel. This partially resolves open questions posed in [DNRS99, BLV03].

The other application (or extension) is that our hash family has the following interesting *universality* properties for correlation intractability, assuming only plain LWE: if any one of a class of hash functions is correlation intractable for all (even inefficiently verifiable) sparse relations, then our family is correlation intractable for all (efficiently verifiable and sufficiently sparse) relations. Remarkably, universality holds even for *multi-input* correlation intractability (namely, when the relation can depend on multiple inputs to the hash function and the corresponding outputs).

We now describe our contributions in more detail.

Correlation Intractability from Fully Homomorphic Encryption

We focus on obtaining correlation intractability for the following class of relations. First, we consider relations R where for every x there is a single y such that $R(x, y)$ holds. We let f denote the function that maps x to the corresponding y that makes $R(x, y)$ hold. That is, $R(x, y) = 1$ iff $y = f(x)$. We say that R is *searchable* in time T if f is computable in time T. We then construct, for each time bound T, a hash function family that is CI with respect to all relations that are searchable in time T. That is:

Theorem 3.1. *If there exists a circular secure fully homomorphic encryption scheme, then for every polynomial time bound $T = T(\lambda)$, every polynomial input size $n = n(\lambda)$ and every constant $\epsilon > 0$, there exists a hash family $\mathcal{H}$ that is correlation intractable for all relations that are searchable in time T, with input size n and output size λ^ϵ.*

We emphasize that efficient searchability is quite different than the notions of efficient relations used in prior work [CCR16, HL18, CCH+18]. Still, we will see that it suffices for our needs. Moreover, our construction is very different from most prior work on correlation intractability: we show that a random key k is indistinguishable from a key k' for which there *do not exist* any x for which $h_k(x) = f(x)$. We call this property "somewhere statistical correlation intractability" in analogy to the notion of "somewhere statistically binding" hash functions of [HW15b]. This statistical property is also what allows us to modify our NIZK argument system to obtain NIZK *proofs* rather than just arguments. See more details in Section 3.1.3

Universal CI. We also show that our *particular* hash function $h(k, x)$ with some fixed time bound T is correlation-intractable for general efficiently verifiable relations of sufficient sparsity, assuming that:

1. There exists *some* hash function $h'(\cdot, \cdot)$ of size T which is correlation-intractable for general (even inefficiently verifiable) relations of sufficiently smaller sparsity.

2. The FHE scheme is semantically secure (we do not rely on circular security for this result).

In addition, our universality argument even extends to the case of *multi-input* correlation intractability, about which very little is currently known.

We note that the flavor of universality demonstrated in this work is very different than other universality results which rely on "Levin's trick" [Lev73]. Specifically, Levin's trick involves guessing the description of a Turing Machine M that securely implements the primitive, and the resulting universal schemes incur a security loss which is exponential in the length of M. Although this is only a constant loss, it is likely to be quite large. In contrast, our universal scheme does not involve guessing a Turing Machine and does not incur the corresponding security loss. In fact, in contrast to Levin's trick, our technique even works in the "non-uniform" setting: if we only start with the premise that there exists a non-uniform constructions of a secure correlation-intractable hash family, then our construction (which is uniform) is still secure (but the security reduction is non-uniform).

Applications to Fiat-Shamir and NIZK

By applying our hash family from Theorem 3.1 to a particular 3-round proof system for graph Hamiltonicity based on [FLS90], we obtain NIZK arguments in the common reference string model from any (circular-secure) fully homomorphic encryption scheme.

Theorem 3.2. *If there exists a circular-secure fully homomorphic encryption scheme, then there exist (adaptively sound) NIZK arguments for* **NP** *in the common reference string model.*

In fact, we prove two different strengthenings of Theorem 3.2: we construct (non-adaptively sound) non-interactive *statistical* zero-knowledge (NISZK) arguments for **NP** in the common *random* string model, and we construct statistically (and adaptively) sound NIZK *proofs* for **NP** in the common reference string model.

Theorem 3.3. *If there exists a circular secure fully homomorphic encryption scheme, then there exist statistically (and adaptively) sound NIZK proofs for* **NP** *in the common reference string model.*

Theorem 3.4. *Suppose that there exists a circular secure fully homomorphic encryption scheme with pseudorandom ciphertexts and public keys. Furthermore, suppose that there exists a lossy public key encryption scheme [KN08, PVW08, BHY09] with uniformly random lossy public keys. Then, there exist (non-adaptively sound) NISZK arguments for* **NP** *in the common random string model.*

The additional hypotheses of Theorem 3.4 are satisfied under LWE. Interestingly, to the best of our knowledge, we did not previously have NISZK argument systems in the common random string model from any standard cryptographic assumption (the [GOS06] NISZK argument system requires a non-random common reference string). Also, we previously did not have any approach toward achieving statistically sound proofs via the Fiat-Shamir heuristic.

Note on Adaptively Sound NISZK: An earlier version of this paper erroniously claimed to construct adaptively sound NISZK arguments; in fact, there are notable barriers to obtaining such a result [Pas13] and we do not prove that our NIZKs can simultaneously satisfy these two properties (see footnote 13 regarding Theorem 3.48). To reiterate, our NIZK arguments are shown to satisfy either adaptive soundness or statistical zero knowledge, but not both simultaneously.

Fiat-Shamir for Trapdoor Σ-Protocols As explained above, our hash family $\mathcal{H}$ can be used to soundly instantiate the Fiat-Shamir heuristic for a particular modification of the 3-round proof system of [FLS90]. More generally, we can apply Fiat-Shamir to "trapdoor Σ-protocols" (see Definition 3.50): roughly speaking, these are 3-message protocols Π in the common reference/random string (CRS) model with the following two properties:

- If the statement x is false, then for every first message $\mathbf{a}$, there is a *unique* challenge $\mathbf{e}$ for which there is an accepting third message $\mathbf{z}$ that results in an accepting transcript $(\mathbf{a}, \mathbf{e}, \mathbf{z})$.

- There is a *trapdoor* τ associated with the CRS that allows us to efficiently compute this "bad challenge" $\mathbf{e}$ from the first message $\mathbf{a}$.

In this language, we modify the [FLS90] 3-round proof system to make it a "trapdoor Σ-protocol" by choosing a commitment scheme that has a commitment public key (which we put in the CRS) for which there exists a trapdoor allowing for extraction. Moreover, we define a generalization called an "instance-dependent trapdoor Σ-protocol" (see Definition 3.51), in which the trapdoor is allowed to depend on the instance x, that also captures the *unmodified* [GMR85] protocol. We prove that our hash family suffices to instantiate Fiat-Shamir for all such protocols. By [DNRS99], this implies that the (parallel repeated) [GMR85] protocol is not zero knowledge.

Corollary 3.5. *Suppose that there exists a circular secure fully homomorphic encryption scheme, and further assume the hardness of quadratic residuosity. Then for any*

$\epsilon > 0$*, the [GMR85] protocol for quadratic residuosity, repeated* λ^ϵ *times in parallel, is not zero knowledge.*

About the Assumption: Circular Secure FHE

We know how to construct leveled FHE under the learning with errors (LWE) assumption [BV11,BGV12,Bra12,GSW13,BV14] which is in turn as hard as worst-case lattice problems [Reg05, BLP+13, PRSD17]. As far as we know, it is reasonable to assume that any of these FHE schemes is circular secure, meaning that if we encrypt the secret key (one bit at a time) then this is indistinguishable from encrypting a dummy message consisting of all 0s. This is a "fully falsifiable" assumption where we only need to assume a standard poly/negligible level of security. In fact, this assumption is needed to perform bootstrapping [Gen09] and is currently the only known approach[1] to get *fully* (non-leveled) homomorphic encryption. Moreover, circular security appears to be a very mild assumption: as far as we know all *natural* encryption schemes that are semantically secure are also circular secure. In fact, it is highly non-trivial to come up with even contrived constructions of semantically secure encryption schemes which are *not* circular secure and we had no such examples until fairly recently with the works of [Rot13, GKW17b, GKW17a, WZ17]. Although we do not know how to prove the circular security of any FHE candidate under LWE directly, we consider circular secure FHE to be a mild, fully falsifiable, lattice-based assumption. Our work achieves the first constructions of correlation-intractable hash functions, instantiations of Fiat-Shamir and NIZKs under such assumptions.

3.1.2 Prior Work on Correlation Intractability and Fiat-Shamir

This work continues a recent line of works [CCR16,KRR17,CCRR18,HL18,CCH+18] focused on constructing correlation-intractable hash families and using them to instantiate the Fiat-Shamir transform in the standard model. Throughout, we consider and compare the following main aspects of the constuctions and assumptions in these

[1] It has also been shown that indistinguishability obfuscation can be used to bootstrap FHE [CLTV15], but there are currently no instantiations of IO from standard assumptions.

works (we consider game-based assumptions):

(a) Our level of familiarity with the assumption

(b) Formal characterictics of the assumption, namely:

  (b.1) The complexity of the algorithm conducting the security game and deciding whether a purported adversary won the game (is it exponential in the security parameter?)

  (b.2) The bound on the allowed success probability of the adversary (is it exponential in the security parameter?)

(c) The class of relations for which correlation intractability is achieved; in particular, whether the hash family is *compact* (namely whether the a single family of functions can withstand relations of arbitrary polynomial size).

The works are described below.

- [CCR16] constructs hash functions that are correlation intractable for all efficiently verifiable relations assuming subexponentially secure indistinguishability obfuscation (IO) [BGI$^+$01, GGH$^+$13] for all circuits as well as input-hiding obfuscation [BCKP14] for all evasive circuits. Both of these assumptions are non-standard; indeed, IO has no constructions from standard assumptions, and input-hiding obfuscation is even less understood. The [CCR16] construction is non-compact: the description size of the hash function depends polynomially on the maximum description size of the relations covered. Using ideas from [HL18], the [CCR16] construction can be used to instantiate the Fiat-Shamir heuristic for specific 3-message protocols of interest in a similar way to what is done in later works.

- [KRR17] (independently of [CCR16]) constructs hash functions that are correlation intractable for *all sparse relations*; in particular, they instantiate the

Fiat-Shamir transform for all constant-round interactive proofs, yielding a construction of NIZK arguments as well as showing that no constant-round public-coin zero knowledge proofs exist. They do so assuming subexponential indistinguishability obfuscation, and in addition a strong variant of point function obfuscation satisfying a form of "fully exponential KDM security." Roughly speaking, this requires that it is fully exponentially hard for a polynomial-time adversary to recover a point x^* given an obfuscation of a program that outputs $y^* = f(x^*)$ on a particular (random) input x^*, *for any* (possibly inefficient) function f.

- [CCRR18] obtains results similar to [KRR17], with significantly simpler and more efficient constructions that avoid obfuscation. Furthermore, their assumptions pertain to security properties of known constructs such as Regev and El-Gamal encryption. Still, their assumptions have the same strong flavors as those of [KRR17]: in particular, they require the existence of an encryption scheme with the property that it is fully exponentially hard to recover the secret key sk given a KDM-encryption $\mathsf{Enc}(\mathsf{sk}, f(\mathsf{sk}))$ for any (possibly inefficient) function f (where Enc is either Regev or El-Gamal encryption).

- [HL18] constructs a correlation-intractable hash family for all relations sampleable in a bounded polynomial time, assuming subexponential IO and exponentially secure one-way functions. This removes the KDM-style assumption (as compared to [KRR17]) but retains the reliance on indistinguishability obfuscation and fully exponential hardness. Their construction is also non-compact. Still, their hash family suffices to instantiate the Fiat-Shamir heuristic for a wide class of 3-message protocols – a strictly broader class than the protocols that we can handle in this work.

- [CCH+18] constructs two correlation-intractable hash families for efficiently sampleable relations: a compact one (i.e., a single poly-size family that covers all poly-size relations) and a non-compact one. Unlike the [CCRR18] constructions, both of these hash families are secure under lattice assumptions that

Reference (Functionality)	Assumes IO?	Exotic Assumption?	Exp-time Challenger?	Exponential Probability?
[CCR16] (Non-Compact, verifiable relations)	Yes	Yes++	No	No
[KRR17] (Compact, all relations)	Yes	Yes	Yes	Yes
[CCRR18] (Compact, all relations)	No	No	Yes	Yes
[HL18] (Non-Compact, sampleable relations)	Yes	No	No	Yes
[CCH$^+$18] (Compact, sampleable relations)	No	No	No	Yes
This work (Non-Compact, searchable relations)	No	No	No	No

Figure 3-1: Constructions of Correlation Intractable Hash Families

are falsifiable in polynomial time (but with exponentially small success probability). The compact family requires a form of circular security, whereas the non-compact one relies on plain search-LWE (albeit with fully exponentially small success probability). Their non-compact family suffices to instantiate Fiat-Shamir for a broad class of protocols similarly to [HL18, CCR16]. Furthermore, their compact scheme suffices for applying the Fiat-Shamir paradigm to the GKR interactive proof [GKR08], thereby obtaining a publicly verifiable succinct non-interactive argument for logspace-uniform NC without assuming indistinguishability obfuscation or non-interactive knowledge extraction from general adversaries.

In Fig. 3-1, we compare key features of the above works. As evident from our description, [KRR17] and all subsequent works have a "fully exponential success probability" barrier: they can only prove security under an assumption that polynomial time adversaries cannot solve some problem with probability significantly better than random guessing. This seems somewhat inherent to the proof technique used in all of these results.

We note that this work is a direct follow-up to [CCH$^+$18]. Indeed, the results and techniques of [CCH$^+$18] were the initial inspiration for this work.

3.1.3 Our Techniques

We give a high-level overview of the proofs of Theorem 3.1, Theorem 3.2, and Corollary 3.5. We begin with our main contribution: a new correlation-intractable hash family.

CI for Efficiently Searchable Relations. We construct a hash-function family $h(k, x)$ with a public hash-key k and input x that satisfies correlation-intractability for all "efficiently searchable relations" with some fixed polynomial time bound T, meaning the following. For any function f having circuit size T, if a polynomial time adversary is given a random k, he cannot find an input x such that $h(k, x) = f(x)$. The output length of the hash function can be as low as λ^ϵ for any $\epsilon > 0$ and the input length can be an arbitrary polynomial in λ. Note that our hash family h only depends on the bound T but not on f; it is correlation intractable for *all* functions f of size T.

At a high level, the idea of the construction is the following. Designing a hash function $h_f(k, x)$ that is correlation intractable for a single function f is trivial: simply define $h_f(k, x) = f(x) + 1$ (or, just flip the last bit of $f(x)$). We will construct a hash function family so that, for *any* f, a random function from the family will look indistinguishable from a hash function that is specifically designed to be correlation intractable with respect to f.

The actual construction is simple:

$$h(k, x) = \mathsf{FHE.Eval}_{\mathsf{pk}}(U_x, \mathsf{ct}), \text{ where } k = (\mathsf{pk}, \mathsf{ct}),\ \mathsf{ct} = \mathsf{FHE.Enc}(\mathsf{pk}, g_0), \text{ and } U_x(g) = g(x). \tag{3.1}$$

That is, the hash function interprets the hash-key $k = (\mathsf{pk}, \mathsf{ct})$ as a public key pk of an FHE scheme, along with a ciphertext ct encrypting some fixed circuit g_0 with input length $|x|$, 1-bit output, and description size m which is related to T (the specific structure of g_0 is unimportant; in particular, it can be the all-zero circuit, i.e. $g_0 = 0^{T'}$ for some T'). The hash function then interprets its input x as the universal circuit $U_x(g) = g(x)$, and homomorphically evaluates $U_x(\cdot)$ over the ciphertext ct. We note that if the FHE scheme in use has pseudorandom public-keys and ciphertexts, we can even choose k as a uniformly random string.

Our proof of security is also simple. Assume that an adversary gets k and is able to find x such that $h(k, x) = f(x)$ with non-negligible probability. We first switch the ciphertext ct in the key k to be an FHE encryption of the circuit $g(x) = \mathsf{Dec}_{\mathsf{sk}}(f(x)) \oplus 1$, where sk is the FHE secret key. In other words, g first computes $f(x)$, then interprets it as an FHE ciphertext of a 1-bit plaintext, decrypts it and outputs the opposite bit.[2] We argue that this change is indistinguishable to the adversary by the security of the FHE; this requires circular security since the circuit g depends on sk.[3] Since the adversary cannot distinguish this change, it still outputs x such that $h(k, x) = f(x)$ with non-negligible probability. So, we have:

$$\begin{aligned} f(x) = h(k,x) &= \mathsf{FHE.Eval}_{\mathsf{pk}}(U_x, \mathsf{ct}) \\ &= \mathsf{FHE.Eval}_{\mathsf{pk}}(U_x, \mathsf{Enc}_{\mathsf{pk}}(\langle \mathsf{Dec}_{\mathsf{sk}}(f(\cdot)) \oplus 1 \rangle), \end{aligned} \tag{3.2}$$

where $U_x(\langle \mathsf{Dec}_{\mathsf{sk}}(f(\cdot)) \oplus 1 \rangle) = \mathsf{Dec}_{\mathsf{sk}}(f(x)) \oplus 1$. However, applying $\mathsf{Dec}_{\mathsf{sk}}(\cdot)$ to both sides of (3.2) we get

$$\mathsf{Dec}_{\mathsf{sk}}(f(x)) = \mathsf{Dec}_{\mathsf{sk}}(\mathsf{FHE.Eval}_{\mathsf{pk}}(U_x, \mathsf{Enc}_{\mathsf{pk}}(\langle \mathsf{Dec}_{\mathsf{sk}}(f(\cdot)) \oplus 1 \rangle))) = \mathsf{Dec}_{\mathsf{sk}}(f(x)) \oplus 1,$$

where the last equality follows by correctness of $\mathsf{FHE.Eval}$. In other words, once we switched ct to be an encryption of g, we ensured that there is no x for which $h(k, x) = f(x)$. This is because we ensure that $h(k, x)$ outputs a ciphertext that is guaranteed to decrypt to a different value than the value v obtained by applying the decryption algorithm to $f(x)$. (We stress that we do not assume any "semantic" meaning to the value v. Indeed, $f(x)$ is in general not a valid ciphertext, so there are

[2] Without loss of generality, we assume that the decryption algorithm always outputs some bit $b \in \{0, 1\}$; e.g., if the decryption algorithm finds a ciphertext to be invalid then it outputs 0.

[3] In slightly more detail, instead of encrypting g directly we separately encrypt f, sk and then homomorphically compute g. This allows us to just rely on circular security (encrypting the secret key itself) rather than key-dependent message security (encrypting functions of the secret key).

no guarantees as to what v might be. Still, it is a well-defined binary value.)

We note that the above proof actually demonstrates a property stronger than plain correlation intractability: For any function f of size T, we can switch the hash-key k to a computationally indistinguishable hash-key $k' = k'_f$ such that the hash function is statistically correlation intractable for f: there does not exist any x such that $h(k', x) = f(x)$. This is reminiscent of the somewhere statistically binding hashing of [HW15b]; we call such hash functions *somewhere statistically correlation intractable.*

In terms of parameters, the output size of the hash function needs to be as large as a single FHE ciphertext encrypting a single bit, which can be set to be as low as λ^ϵ for any $\epsilon > 0$. On the other hand, the size of the key $k = (\mathsf{pk}, \mathsf{ct})$ and the time to evaluate $h(k, \cdot)$ depend on T; this is because ct needs to be large enough to encrypt f and U_x needs to be large enough to evaluate f.[4]

NIZKs for NP via Fiat-Shamir. We show that CI for efficiently searchable relations is sufficient to instantiate the Fiat-Shamir heuristic for a particular Σ-protocol (i.e. 3 round public-coin protocol with "special soundness") and get NIZK arguments for NP. This follows the general framework explored in [HL18] and [CCH+18].

We take the Σ-protocol of [FLS90] for showing that a graph G has a Hamiltonian cycle. The prover sends a commitment to a random cycle graph C. The verifier sends a challenge bit. If the bit is 0, the prover decommits to the entire graph and the verifier checks that it is indeed a cycle. If the bit is 1, the prover sends a random permutation of G which maps the Hamiltonian cycle in G to C along with the opening of all the non-edges of the permuted G. We amplify soundness via parallel repetition. Borrowing an idea from [HL18], we use a public-key encryption scheme to implement the commitment, where the public key pk is in the CRS.

The above Σ-protocol has the following property. If the statement is false, then given the prover's first message a, there is a unique "bad" challenge e that for which

[4] If f has a succinct description as a Turing Machine that runs in time T, we can make the key k shorter than T by having ct encrypt the Turing Machine description of f and letting U_x be a universal circuit which takes as input a Turing Machine and runs it for T steps. Still, the time to evaluate $h(k, \cdot)$ depends on the run-time T.

a valid response z exists. Furthermore, this "bad" challenge would be efficiently computable if we had all the committed values. Since we use a public-key encryption scheme as a commitment, we can extract the committed values from the commitment a using the encryption secret key sk. Combining the above, given the encryption secret key sk, there is an efficiently computable function $f_{\mathsf{sk}}(a)$ which maps the prover's first message a to the unique "bad" challenge e that has a valid response. The size of the function f_{sk} is bounded by some bound T.

We show that, if we apply the Fiat-Shamir heuristic to the above protocol and use a hash function which is correlation-intractable for all "efficiently searchable relations" with the bound T, we get a NIZK argument. Recall that the Fiat-Shamir heuristic adds a hash key k to the CRS and requires the prover to come up with a valid protocol transcript (a, e, z) where $e = h(k, a)$. Since the hash function is correlation-intractable for all "efficiently searchable relations" with the bound T, it is in particular correlation-intractable for f_{sk}. This means that an efficient prover cannot come up with a value a such that $h(k, a) = f_{\mathsf{sk}}(a)$. But if the statement is false, then the only way a proof (a, e, z) can be valid is if $h(k, a) = e = f_{\mathsf{sk}}(a)$. Therefore, the prover cannot come up with any valid proofs and we have soundness. The zero-knowledge (ZK) property of the NIZK follows from the honest-verifier zero-knowledge property of the Σ-protocol.[5] We elaborate that in the above argument, we only need the hash function to be correlation-intractable for a particular function f_{sk} but since sk is secret (releasing it would break zero-knowledge) we rely on the fact that we can choose the hash key k in a way that does not reveal sk.

We obtain a statistically sound NIZK *proof* system by using a "statistically correlation intractable" hash key $k_{f_{\mathsf{sk}}}$ instead of a plain hash key k. This makes use the fact that the function f_{sk} depends on a trapdoor to the 3-message CRS but not on the instance x. Now, when we argue zero-knowledge, we make use of the computational property that $k_{f_{\mathsf{sk}}}$ is indistinguishable from a random k which does not depend on sk.

Finally, if we use a "lossy encryption" scheme to implement the commitments,

[5] For this to work, we also need the hash function to be 1-universal, meaning that for any a the value $h(k, a)$ is uniformly random over the choice of k. We show that 1-universality can be generically added to any correlation-intractable hash function.

we obtain statistical zero-knowledge. In this variant, we can also make the CRS truly random assuming that we have FHE where the public-keys and ciphertexts are pseudorandom (implied by circular LWE) and that we have "lossy encryption" with random public keys (implied by LWE).

Fiat-Shamir for the [GMR85] Protocol. By a very similar argument to our NIZK construction, we show that the [GMR85] Quadratic Residuosity protocol is not zero knowledge when repeated a large number of times in parallel, assuming the existence of correlation-intractable hash functions for efficiently searchable relations. This takes advantage of the aforementioned [DNRS99] result that if there exists a hash function that suffices for the Fiat-Shamir transform for a protocol Π (for a language $L \notin \mathsf{BPP}$), then Π cannot be zero knowledge. In this overview, we use [GMR85] as an example for the general notion of an "instance-dependent trapdoor Σ-protocol" that we introduce.

Recall the [GMR85] protocol: in order to prove that a number y is a quadratic residue modulo a composite number $N = pq$, the prover sends to the verifier a random square $a = r^2$ modulo N; the verifier sends a random bit e to the prover, at which point the prover reveals a square root of $a \cdot y^e$ (either r or rx for some square root x of y).

To show that our hash family suffices to instantiate Fiat-Shamir for the [GMR85] protocol (repeated in parallel), we note that the soundness of Fiat-Shamir for this protocol follows from correlation intractability for the function

$$f_N(\mathbf{a}) = \mathbf{e} := (e_i = QR(N, a_i))_i,$$

where $QR(N, a) = 1$ if and only if a_i is a square modulo N. This function f_N simply computes, for every first message $\mathbf{a}$ in the (parallel repeated) [GMR85] protocol, the unique challenge $\mathbf{e} \in \{0,1\}^t$ that a cheating prover has any hope of being able to win on (provided that the instance y is not a quadratic residue).

While f_N is not efficiently computable as a function of $(N, \mathbf{a})$, it *is* efficiently computable given the factorization $N = pq$ as non-uniform advice, so we can show that our hash family is correlation intractable for every f_N and hence demonstrates our claimed result.

At a high level, the [GMR85] protocol is similar to our modified [FLS90] protocol in that no-instances (N, y) have an associated "bad challenge function" f_N and there is a trapdoor (p, q) making f_N efficiently computable. However, in this case, the trapdoor depends on the instance (N, y) (as opposed to in the [FLS90] modification, where it only depends on the CRS). This motivates our definition of an "instance-dependent trapdoor Σ-protocol" in Section 3.6.

Universal CI. We also show that our *particular* hash function $h(k, x)$ described in (3.1) with some fixed time bound T is correlation-intractable for general efficiently decidable relations of sufficient sparsity, assuming that:

1. There exists *some* hash function $h'(\cdot, \cdot)$ of description size T which is correlation-intractable for general (even inefficient) relations of sufficient (necessarily larger) sparsity.

2. The FHE scheme is semantically secure (we do not rely on circular security for this result).

To see why our construction is "universal", assume for contradiction that there is some sufficiently sparse and efficiently computable relation R as well as an adversary that, given the key $k = (\mathsf{pk}, \mathsf{ct})$ of our hash function, computes x such that $(x, h(k, x)) \in R$ with non-negligible probability. We first switch ct to be an encryption of the correlation-intractable hash function $h'(k', \cdot)$ for a random k'. By the semantic security of the encryption, this is indistinguishable and therefore the adversary still produces x such that $(x, h(k, x)) \in R$ with non-negligible probability. Since $h'(k', \cdot)$ is correlation-intractable for all sufficiently sparse relations, it is in particular correlation-intractable for the (inefficient) relation[6]:

$$R^*_{\mathsf{sk}} = \{(x, z) : \exists y \text{ such that } (x, y) \in R \text{ and } z = \mathsf{Dec}(\mathsf{sk}, y)\},$$

[6] For this argument to work, parameters must be set so that this relation is still sparse.

But if $(x, y = h(k,x)) \in R$ then $y = \mathsf{Enc}_{\mathsf{pk}}(h'(k',x))$ and therefore, for $z = \mathsf{Dec}(\mathsf{sk}, y)$, we have $(x, z = h'(k',x)) \in R^*_{\mathsf{sk}}$. So the adversary also breaks the correlation intractability of $h'(k',\cdot)$ with respect to the relation R^*_{sk}, which should be impossible.

3.1.4 Subsequent Work

Following this work, Peikert and Shiehian [PS19] give a beautiful construction of a (somewhere statistically) correlation intractable hash family from the *plain* LWE assumption, yielding NIZKs from plain LWE. We briefly provide an interpretation of their construction in light of our high-level paradigm.

Recall that for our hash family is defined by having an FHE encryption $\mathsf{Enc}_{\mathsf{pk}}(f)$ of a function f in the hash key, and the hash function evaluation on input x consists of homomorphically evaluating the function $U_x(f) = \mathsf{Dec}_{\mathsf{sk}}(f(x)) \oplus 1$ under FHE. Note that the function $U_x(f)$ depends on sk. In our work, this evaluation procedure is implemented by releasing an encryption $\mathsf{FHE.Enc}(\mathsf{sk})$ of the FHE secret key sk as part of the hash key, and this forces us to rely on circular secure FHE to prove security of our hash family. At a high level, the work of [PS19] cleverly shows that one can homomorphically evaluate $U_x(f)$ directly without needing to release an encryption of the secret key! This is done by switching between two different encryption schemes: the "input" ciphertext is an encryption of f under the GSW FHE scheme [GSW13], while the "output" ciphertext is tantamount to an encryption of $U_x(f) = \mathsf{Dec}_{\mathsf{sk}}(f(x)) \oplus 1$ under the Regev encryption scheme (with the same secret key used for both schemes).

At a high level, given a GSW encryption of f, it is possible to compute a GSW encryption of $f(x)$ using the GSW homomorphic evaluation procedure. One can then "downgrade" the GSW encryption of $f(x)$ to a Regev ciphertext and, in doing so, incorporate the secret key into the computation (for some intuition as to why this is possible, note that Regev encryption is circular secure under the plain LWE assumption and therefore we can get Regev encryptions of the secret key for free). However, since Regev encryption is no longer "fully homomorphic" but only "additively homomorphic", after downgrading to a Regev encryption, only linear functions can be evaluated. Luckily, this suffices to perform a Regev decryption of $f(x)$ and therefore one can homomorphically derive a Regev ciphertext encrypting $U_x(f)$.

There is a caveat with the above due to the fact that Regev decryption also involves rounding, which is non-linear. To get around this, one can think of a "noisy Regev" variant that operates over the message space $\mathbb{Z}_q$, and decryption does not perform rounding, but correctness is only approximate – the decrypted value is close to the encrypted one. One can then define $U_x(f) = \mathsf{Dec}_{\mathsf{sk}}(f(x)) + \lfloor q/2 \rfloor$ where Dec is the linear "noisy Regev" decryption procedure. Using the above template, given a GSW encryption of f, one can compute an encryption of $U_x(f)$ under the "noisy Regev" scheme. This still ensures that the hash of x, which is a "noisy Regev" encryption of $U_x(f)$, cannot be equal to $f(x)$ since they decrypt to values in $\mathbb{Z}_q$ that are far from each other.

3.1.5 Organization

The remainder of the paper is organized as follows. In Section 3.2, we recall basic preliminaries, and in Section 3.3, we define correlation intractability [CGH98] and the specific variants focused on in this work. In Section 3.4, we present our main constructions of correlation intractable hash families from fully homomorphic encryption. Finally, we apply these hash families in Section 3.5 to obtain our main results (Theorem 3.2 and its extensions), and in Section 3.6 to obtain our most general Fiat-Shamir instantiation.

3.2 Preliminaries

We say that a function $\mu(\lambda)$ is *negligible* if $\mu(\lambda) = O(\lambda^{-c})$ for every constant c, and that two distribution ensembles $X = \{X_\lambda\}$ and $Y = \{Y_\lambda\}$ are computationally indistinguishable ($X \approx_c Y$) if for all polynomial-sized circuit ensembles $\{\mathcal{A}_\lambda\}$,

$$\left| \Pr\left[\mathcal{A}_\lambda(X_\lambda) = 1\right] - \Pr\left[\mathcal{A}_n(Y_\lambda) = 1\right] \right| = \text{negl}(\lambda).$$

3.2.1 (Lossy) Public Key Encryption

Definition 3.6 (Public Key Encryption)**.** *A public-key encryption scheme* $\mathsf{PKE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *consists of three p.p.t. algorithms:*

- $\mathsf{Gen}(1^\lambda)$ *takes as input the security parameter and outputs a public key* pk *and a secret key* sk*.*
- $\mathsf{Enc}(\mathsf{pk}, m)$ *takes as input the public key and a bit*[7] $m \in \{0,1\}$*; it outputs a ciphertext* ct*.*
- $\mathsf{Dec}(\mathsf{sk}, \mathsf{ct})$ *takes as input the secret key and a ciphertext* ct*; it outputs a message* m'*.*

PKE *must furthermore satisfy the following properties.*

- ***Correctness****: For all* λ*, all* $m \in \{0,1\}$*, and all* $(\mathsf{pk}, \mathsf{sk})$ *in the support of* 1^λ*, it holds with probability* 1 *that* $\mathsf{Dec}(\mathsf{sk}, \mathsf{Enc}(\mathsf{pk}, m)) = m$*.*
- ***Semantic Security****: The distribution ensembles* $\{(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^\lambda) : (\mathsf{pk}, \mathsf{Enc}(\mathsf{pk}, 0))\} \approx_c \{(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^\lambda) : (\mathsf{pk}, \mathsf{Enc}(\mathsf{pk}, 1))\}$ *are computationally indistinguishable.*

We say that a public key encryption scheme PKE *has* pseudorandom ciphertexts *if the distribution ensembles* $\{(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^\lambda) : (\mathsf{pk}, \mathsf{Enc}(\mathsf{pk}, 0))\} \approx_c \{(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^\lambda), u \leftarrow U_{|\mathsf{Enc}(\mathsf{pk},0)|} : (\mathsf{pk}, u)\}$*, where* $\approx_c$ *denotes computational indistinguishability, and* PKE *has* pseudorandom public keys *if a public key* pk *sampled according to* $\mathsf{Gen}(1^\lambda)$ *is computationally pseudorandom.*

Definition 3.7 (Lossy PKE)**.** *A public-key encryption scheme* PKE *is said to be* lossy *[KN08, PVW08, BHY09] if there exists a* fake key generation algorithm $\mathsf{FakeGen}$ *such that:*

- *The (randomized) output* $\widetilde{\mathsf{pk}}$ *of* $\mathsf{FakeGen}(1^\lambda)$ *is computationally indistinguishable from a public key* pk *sampled by* $\mathsf{Gen}(1^\lambda)$*.*

[7] As usual, this extends naturally to encrypting many-bit plaintexts.

- *Encryption under fake keys is statistically hiding. That is,*

$$\{(\widetilde{\mathsf{pk}}, \mathsf{Enc}(\widetilde{\mathsf{pk}}, 0))\} \approx_s \{(\widetilde{\mathsf{pk}}, \mathsf{Enc}(\widetilde{\mathsf{pk}}, 1))\},$$

  where $\widetilde{\mathsf{pk}} \leftarrow \mathsf{FakeGen}(1^\lambda)$ *and* $\approx_s$ *denotes statistical indistinguishability.*

We say that PKE *is* lossy with uniformly random lossy public keys *if (in addition)* FakeGen *outputs a uniformly random string.*

In this work, we make use of the fact that public-key Regev encryption [Reg05] is lossy with uniformly random lossy public keys under the LWE assumption.

3.2.2 Fully Homomorphic Encryption and Circular Security

Definition 3.8. *A* fully homomorphic encryption scheme $\mathsf{FHE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval})$ *consists of four p.p.t. algorithms such that* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *is a public key encryption scheme, and:*

- $\mathsf{Eval}(\mathsf{pk}, f, \mathsf{ct}_1, \ldots, \mathsf{ct}_n)$ *takes as input the public key, a function* f *(represented by a boolean circuit), and a vector of ciphertexts* $(\mathsf{ct}_1, \ldots, \mathsf{ct}_n)$*; it outputs another ciphertext* ct'*, which has size that is polynomial in* λ *(and, without loss of generality, linear in the output length of* f*).*

- *For any* $(\mathsf{pk}, \mathsf{sk}) \leftarrow (\mathsf{Gen}(1^\lambda))$*, any* $m_1, \ldots, m_n \in \{0,1\}$*, and any circuit* $C : \{0,1\}^n \to \{0,1\}$*, it holds with probability* 1 *that*

$$\mathsf{Dec}\Big(\mathsf{sk}, \mathsf{Eval}\Big(\mathsf{pk}, C, \mathsf{Enc}(\mathsf{pk}, m_1), \ldots, \mathsf{Enc}(\mathsf{pk}, m_n)\Big)\Big) = C(m_1, \ldots, m_n).$$

Definition 3.9. *A* leveled fully homomorphic encryption scheme $\mathsf{FHE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval})$ *satisfies the same syntax, correctness, and security properties of a FHE scheme, except that*

- $\mathsf{Gen}(1^\lambda, 1^d)$ *takes as additional input a* circuit depth *d.*

- *Homomorphic evaluation correctness is only guaranteed to hold for circuits of depth at most d.*

- *Ciphertexts output by* $\mathsf{Enc}(\mathsf{pk}, m)$ *and* $\mathsf{Eval}(\mathsf{pk}, f, \mathsf{ct})$ *have size that are polynomial in λ (and the output length of f), independent of d.*

- *The decryption algorithm* $\mathsf{Dec}(\mathsf{sk}, \mathsf{ct})$ *has a fixed* $\mathsf{poly}(\lambda)$ *depth (independent of d).*

Leveled fully homomorphic encryption schemes are known to exist from the learning with errors (LWE) assumption [BV11,BGV12,Bra12,GSW13,BV14]. Fully homomorphic encryption schemes are known to exist using Gentry's bootstrapping technique [Gen09], which requires making a circular security assumption on an LWE-based encryption scheme. In this work, we consider the following variant of circular security.

Definition 3.10. *A public key encryption scheme* PKE *is said to be* circular secure *if* $\{(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^\lambda) : (\mathsf{pk}, \mathsf{Enc}(\mathsf{pk}, 0^{|\mathsf{sk}|}))\} \approx_c \{(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^\lambda) : (\mathsf{pk}, \mathsf{Enc}(\mathsf{pk}, \mathsf{sk}))\}$.

3.2.3 Non-Interactive Zero Knowledge Arguments (and Proofs)

The following preliminaries are taken (with edits) from [CCH+18].

Definition 3.11. *A* non-interactive zero knowledge (NIZK) argument system Π *for an* **NP** *relation R consists of three ppt algorithms* (Setup, P, V) *with the following syntax.*

- $\mathsf{Setup}(1^n, 1^\lambda)$ *takes as input a statement length n and a security parameter λ. It outputs a common reference string* crs*.*

- $P(\mathsf{crs}, x, w)$ *takes as input the common reference string, as well as x and w such that $(x, w) \in R$. It outputs a proof π.*

- $V(\mathsf{crs}, x, \pi)$ *takes as input the common reference string, a statement x, and a proof π. It outputs a bit b. If $b = 1$, we say that V* accepts*, and otherwise we say that V* rejects*.*

The proof system Π *must satisfy the following requirements for every polynomial function* $n = n(\lambda)$*. Recall that* $\mathcal{L}(R)$ *denotes the language* $\{x : \exists w \text{ s.t. } (x, w) \in R\}$ *and* R_n *denotes the set* $R \cap (\{0,1\}^n \times \{0,1\}^*)$*.*

- **Completeness.** *For every* $(x, w) \in R$*, it holds with probability* 1 *that* $V(\mathsf{crs}, x, \pi) = 1$ *in the probability space defined by sampling* $\mathsf{crs} \leftarrow \mathsf{Setup}(1^{|x|}, 1^\lambda)$ *and* $\pi \leftarrow P(\mathsf{crs}, x, w)$*.*

- **Soundness.** *For every* $\Big\{x_n \in \{0,1\}^n \setminus \mathcal{L}(R)\Big\}$ *and every polynomial size* $P^* = \{P^*_\lambda\}$*, there is a negligible function* ν *such that*

$$\Pr_{\substack{\mathsf{crs} \leftarrow \mathsf{Setup}(1^n, 1^\lambda) \\ \pi \leftarrow P^*_\lambda(\mathsf{crs})}} \Big[V(\mathsf{crs}, x_n, \pi) = 1\Big] \leq \nu(\lambda).$$

- **Zero Knowledge.** *There is a ppt simulator* Sim *such that for every ensemble* $\Big\{(x_n, w_n) \in R_n\Big\}$*, the distribution ensembles*

$$\Big\{\Big(\mathsf{crs}_\lambda, P(\mathsf{crs}_n, x_n, w_n)\Big)\Big\}_\lambda$$

  and

$$\Big\{\mathsf{Sim}(x_n, 1^\lambda))\Big\}_\lambda$$

  are computationally indistinguishable in the probability space defined by sampling $\mathsf{crs}_\lambda \leftarrow \mathsf{Setup}(1^n, 1^\lambda)$ *(and evaluating* P *and* Sim *with independent and uniform randomness).*

  If the distributions are statistically indistinguishable*, then* Π *is said to be* statistically zero knowledge.

A NIZK argument system can also satisfy various stronger properties. We list some important variants below.

- **"Common Random String"**: A NIZK argument system in the common *random* string model is a NIZK argument system Π such that $\mathsf{Setup}(1^n, 1^\lambda)$ simply samples and outputs a uniformly random string.

- **Adaptive Soundness**: Π is adaptively sound if for every polynomial size algorithm $P^* = \{P^*_\lambda\}$, there is a negligible function ν such that for all λ,

$$\Pr_{\substack{\mathsf{crs} \leftarrow \mathsf{Setup}(1^n, 1^\lambda) \\ (x,\pi) := P^*_\lambda(\mathsf{crs})}}[x \notin \mathcal{L}(R) \wedge V(\mathsf{crs}, x, \pi) = 1] \leq \nu(\lambda).$$

- **Statistical Soundness:** Π is statistically sound if there is a negligible function ν such that for all λ,

$$\Pr_{\mathsf{crs} \leftarrow \mathsf{Setup}(1^n, 1^\lambda)}[\exists (x, \pi) \text{ such that } x \notin \mathcal{L}(R) \wedge V(\mathsf{crs}, x, \pi) = 1] \leq \nu(\lambda).$$

  A NIZK argument system satisfying statistical soundness is called a NIZK proof system.

- **Multi-Theorem Zero Knowledge**: Π is multi-theorem zero knowledge if for every polynomial function $p(\lambda)$, there is a p.p.t. simulator Sim such that for every ensemble $\{(x_i^{(n(\lambda))}, w_i^{(n(\lambda))})_{i=1}^{p(\lambda)}\}$, the distribution ensembles

$$\left\{ \left(\mathsf{crs}_\lambda, P(\mathsf{crs}_\lambda, x_i^{(n)}, w_i^{(n)}) \right)_{i=1}^{p(\lambda)} \right\}_\lambda$$

  and

$$\left\{ \mathsf{Sim}(x_1, \ldots, x_{p(\lambda)})) \right\}_\lambda$$

  are computationally indistinguishable. [FLS90] showed a generic transformation from a NIZK proof or argument system to one satisfying multi-theorem zero knowledge. This transformation preserves computational zero knowledge in the common random string model and statistical zero knowledge in the common reference string model.

- **Adaptive Zero Knowledge**: Π is adaptive zero knowledge if for every p.p.t. verifier V^*, there is a p.p.t. simulator Sim such that the following distribution ensembles are computationally indistinguishable:

$$\left\{\mathsf{crs} \leftarrow \mathsf{Setup}(1^n, 1^\lambda), (x, w, \mathsf{aux}) \leftarrow V^*(\mathsf{crs}) : (\mathsf{crs}, P(\mathsf{crs}, x, w), \mathsf{aux})\right\}$$

and

$$\left\{\mathsf{Sim}(1^n, 1^\lambda)\right\}.$$

This can (analogously to above) be extended to a definition of **adaptive multi-theorem zero knowledge**, which can be obtained generically from adaptive zero-knowledge by [FLS90].

3.3 Somewhere Statistically Correlation Intractable Hash Families

In this section, we recall the notion of correlation intractability [CGH98], which is a particular security property associated to a hash family $\mathcal{H}$. We then introduce a new strengthening of this definition, which we call "somewhere statistical correlation intractability" by analogy to the "somewhere statistically binding" hash functions of [HW15b].

We also define new classes of relations – "efficiently searchable" and "efficiently enumerable" relations – for which we later (1) achieve correlation intractability ("somewhere statistical," in the case of efficiently searchable relations), and (2) obtain applications of interest.

Definition 3.12. *For a pair of efficiently computable functions* $(n(\cdot), m(\cdot))$*, a* hash family *with input length* n *and output length* m *is a collection* $\mathcal{H} = \{h_\lambda : \{0,1\}^{s(\lambda)} \times \{0,1\}^{n(\lambda)} \to \{0,1\}^{m(\lambda)}\}_{\lambda \in \mathbb{N}}$ *of keyed hash functions, along with a pair of p.p.t. algorithms:*

- $\mathcal{H}.\mathsf{Gen}(1^\lambda)$ *outputs a hash key* $k \in \{0,1\}^{s(\lambda)}$.

- $\mathcal{H}.\mathsf{Hash}(k, x)$ *computes the function* $h_\lambda(k, x)$*. We may use the notation* $h(k, x)$ *to denote hash evaluation when the hash family is clear from context.*

We cay that $\mathcal{H}$ *is* public-coin[8] *if* $\mathcal{H}.\mathsf{Gen}$ *outputs a uniformly random string* $k \leftarrow \{0,1\}^{s(\lambda)}$.

Definition 3.13 (Correlation Intractability)**.** *For a given relation ensemble* $R = \{R_\lambda \subseteq \{0,1\}^{n(\lambda)} \times \{0,1\}^{m(\lambda)}\}$*, a hash family* $\mathcal{H} = \{h_\lambda : \{0,1\}^{s(\lambda)} \times \{0,1\}^{n(\lambda)} \to \{0,1\}^{m(\lambda)}\}$ *is said to be* R-correlation intractable with security (s, δ) *if for every* s*-size* $\mathcal{A} = \{\mathcal{A}_\lambda\}$,

$$\Pr_{\substack{k \leftarrow \mathcal{H}.\mathsf{Gen}(1^\lambda) \\ x \leftarrow \mathcal{A}(k)}} \Big[\Big(x, h(k, x)\Big) \in R\Big] = O(\delta(\lambda)).$$

We say that $\mathcal{H}$ *is* R-correlation intractable *if it is* $(\lambda^c, \frac{1}{\lambda^c})$*-correlation intractable for all* $c > 1$.

If $\mathcal{R}$ *is a collection of relation ensembles, then* $\mathcal{H}$ *is said to be* uniformly $\mathcal{R}$-correlation intractable *if for every polynomial-size* $\mathcal{A}$*, there exists a function* $\nu(\lambda) = \mathrm{negl}(\lambda)$ *such that for every* $R \in \mathcal{R}$,

$$\Pr_{\substack{k \leftarrow \mathcal{H}.\mathsf{Gen}(1^\lambda) \\ x \leftarrow \mathcal{A}(k)}} \Big[(x, h(k, x)) \in R\Big] \le \nu(\lambda).$$

As noted in [CGH98], a random oracle (typically thought of as an "ideal hash function" [BR93]) behaves like an R-correlation intractable for all *sparse* relations R.

Definition 3.14 (Sparsity)**.** *For any relation ensemble* $R = \{R_\lambda \subseteq \{0,1\}^{n(\lambda)} \times \{0,1\}^{m(\lambda)}\}$*, we say that* R *is* $\rho(\cdot)$-sparse *if for* $\lambda \in \mathbb{N}$ *and any* $x \in \{0,1\}^n$,

$$\Pr_{y \leftarrow \{0,1\}^m} \Big[(x, y) \in R\Big] \le \rho(\lambda).$$

When ρ *is a negligible function, we say that* R *is* sparse*.*

[8] Sometimes "public-coin" hash families are defined to be hash families whose security properties hold even when the adversary is given the random coins used to sample $k \leftarrow \mathcal{H}.\mathsf{Gen}(1^\lambda)$. For our purposes (e.g. ignoring compactness), this definition is equivalent to ours.

We now introduce our new notion of "somewhere statistical correlation intractability."

Definition 3.15 (Somewhere Statistical Correlation Intractability)**.** *Given a collection $\mathcal{R}$ of relation ensembles, we say that a hash family $\mathcal{H}$ is* somewhere statistically correlation intractable *with respect to $\mathcal{R}$ if there is an additional key generation algorithm* $\mathsf{StatGen}$ *with the following properties.*

- ***Syntax:*** $\mathsf{StatGen}(1^\lambda, \mathsf{aux}_\lambda)$ *takes as input the security parameter λ as well as an auxiliary input aux_λ. It outputs a hash key k.*
- ***Security:*** *For any relation ensemble $R \in \mathcal{R}$, there* exists *an auxiliary input ensemble* aux *such that the following two properties hold.*
  - ***Key Indistiguishability:*** *An honestly generated key $k \leftarrow \mathcal{H}.\mathsf{Gen}(1^\lambda)$ is computationally indistinguishable from a fake key $k \leftarrow \mathcal{H}.\mathsf{StatGen}(1^\lambda, \mathsf{aux}_\lambda)$.*
  - ***Statistical Correlation Intractability:***

    $$\Pr_{k \leftarrow \mathcal{H}.\mathsf{StatGen}(1^\lambda, \mathsf{aux}_\lambda)} \left[\exists x \in \{0,1\}^{n(\lambda)} : (x, h(k,x)) \in R_\lambda \right] = \mathrm{negl}(\lambda).$$

    That is, with high probability over the choice of $k \leftarrow \mathsf{StatGen}(1^\lambda, \mathsf{aux}_\lambda)$, input-output pairs satisfying R_λ do not exist.

Remark 3.16. *By a simple hybrid argument, somewhere statistical correlation intractability with respect to a family of relations $\mathcal{R}$ implies (ordinary) correlation intractability for $\mathcal{R}$.*

3.3.1 Efficiently Searchable Relations

In this work, we focus further on achieving (somewhere statistical) correlation intractability for relations R with a *unique* output $y = f(x)$ associated to each input x, and such that $y = f(x)$ is an efficiently computable function of x.

Definition 3.17 (Unique Output Relation)**.** *We say that a relation R is a* unique output relation *if for every input x, there exists at most one output y such that $(x, y) \in R$.*

Remark 3.18. *When restricted to the case of unique output relations, correlation intractable hash functions for output length m immediately imply the existence of correlation intractable hash functions for any output length $m' > m$ (by appending zeros).*

Definition 3.19 (Efficiently Searchable Relation)**.** *We say that a (necessarily unique-output) relation ensemble R is* searchable in (non-uniform) time T *if there exists a function $f = f_R : \{0,1\}^* \to \{0,1\}^*$ computable in (non-uniform) time T such that for any input x, if $(x, y) \in R$ then $y = f(x)$; that is, $f(x)$ is the unique y such that $(x, y) \in R$, provided that such a y exists. We say that R is* efficiently searchable *if it is searchable in time $\mathsf{poly}(n)$.*

We now relate our notion of efficient searchability to that of *efficient sampleability* [HL18,CCH+18]. Efficiently sampleable relations R are not necessarily unique-output, but it is possible to sample, given an input x, an (approximately) uniformly random y subject to the condition $(x, y) \in R$. Correlation intractability for these relations is not required in order for our Fiat-Shamir applications, but as noted below, we obtain it for sufficiently sparse relations without loss of generality. In particular, if the relation has sparsity p, we obtain it with a security loss of $p \cdot 2^m$.

Definition 3.20 (Efficiently (Approximately) Sampleable Relation)**.** *We say that a relation R is* sampleable in (non-uniform) time T *if there exists a (non-uniform) time T algorithm $\mathsf{Samp}(x; r)$ and a polynomial $q(\cdot)$ such that for any $(x^*, y^*) \in R$,*

$$\Pr_r\Big[\mathsf{Samp}(x^*; r) = y^*\Big] \geq \frac{1}{q(\lambda)}\Big|\{y \in \{0,1\}^m : (x, y) \in R\}\Big|^{-1}.$$

Lemma 3.21. *Suppose that a hash family $\mathcal{H}$ is δ-correlation intractable for all relations* searchable *in time T. Then, it is also $\delta p 2^m$-correlation intractable for all p-sparse relations* samplable *in time T.*

Proof. Let R denote a relation that is sampleable in time T with approximation factor $q(\lambda)$, and let $\mathsf{Samp}(x; r)$ denote a sampling algorithm for R. Then, for every fixed r, the relation

$$R_r = \{(x, \mathsf{Samp}(x; r))\}$$

is searchable in time T. Moreover, if some adversary $\mathcal{A}$ breaks the R-correlation intractability of a hash family $\mathcal{H}$ with probability δ', then by an averaging argument, $\mathcal{A}$ breaks the R_r-correlation intractability of $\mathcal{H}$ with probability $\frac{\delta'}{q(\lambda)p2^m}$ for some choice of randomness r. □

In particular, this shows that CI for efficiently searchable relations directly implies CI for efficiently sampleable relations for which every input x has *at most polynomially many outputs* y for which $(x, y) \in R$. We call such relations efficiently enumerable, because this is equivalent to the existence of an efficient algorithm that enumerates all "bad outputs" y for a given input x.

3.3.2 Programmability

As previously discussed, correlation intractability is useful in proving the *soundness* of the Fiat-Shamir transform for certain proof systems, as seen in Section 3.5 and Section 3.6. Since we hope to use our correlation intractable hash families to build NIZK arguments (which in particular must also be *zero knowledge*), we would like to have correlation intractable hash families satisfying a weak notion of *programmability.*

Definition 3.22. *We say that a hash family $\mathcal{H}$ is* 1-universal *if for any λ, input $x \in \{0,1\}^{n(\lambda)}$, and output $y \in \{0,1\}^{m(\lambda)}$, we have that*

$$\Pr_{k \leftarrow \mathcal{H}.\mathsf{Gen}(1^\lambda)}[h(k, x) = y] = 2^{-m}.$$

We say that a hash family $\mathcal{H}$ is programmable *if it is 1-universal, and if there exists an efficient sampling algorithm $\mathsf{Samp}(1^\lambda, x, y)$ that samples from the conditional distribution $k \leftarrow \mathcal{H}.\mathsf{Gen}(1^\lambda) \mid h(k, x) = y$.*

We describe a simple transformation showing that for reasonable classes of relations (including efficiently searchable relations), programmability can be obtained without loss of generality.

Construction 3.23. *Let $\mathcal{H}$ be any hash family. We define the programmable variant $\mathcal{H}' = \mathcal{H}^{\text{prog}}$ of $\mathcal{H}$ as follows:*

- $\mathcal{H}'.\mathsf{Gen}(1^\lambda)$ *calls* $k \leftarrow \mathcal{H}.\mathsf{Gen}(1^\lambda)$*, samples a uniformly random* $\alpha \leftarrow \{0,1\}^m$*, and outputs* (k, α)*.*
- $\mathcal{H}'.\mathsf{Hash}((k,\alpha), x)$ *outputs* $\mathcal{H}.\mathsf{Hash}(k,x) \oplus \alpha$*.*

We first remark that $\mathcal{H}'$ is evidently programmable: 1-universality follows from the randomness of α, and the algorithm $\mathsf{Samp}(1^\lambda, x, y)$ calls $k \leftarrow \mathcal{H}.\mathsf{Gen}(1^\lambda)$ and outputs $(k, \mathcal{H}.\mathsf{Hash}(k,x) \oplus y)$.

Moreover, we note that if $\mathcal{H}'$ directly inherits correlation intractability properties from $\mathcal{H}$.

Remark 3.24. *For any relation class $\mathcal{R}$, if $\mathcal{H}$ is (somewhere statistically) correlation intractable for the class of relations*

$$\{R^*_\alpha = \{(x,y) : (x, y \oplus \alpha) \in R\}\}_{R \in \mathcal{R}},$$

then $\mathcal{H}'$ is (somewhere statistically) correlation intractable for $\mathcal{R}$.

3.4 Correlation Intractability via Fully Homomorphic Encryption

In this section, we describe a new candidate correlation intractable hash family $\mathcal{H}$ that can be based on any fully homomorphic encryption scheme. We then prove that $\mathcal{H}$ satisfies various notions of correlation intractability under different assumptions. Namely, we show:

- One variant of our hash family is (somewhere statistically) correlation intractable for efficiently searchable relations assuming that the FHE scheme is circular secure.

- Another variant of our hash family is "universal" for correlation intractable hash families (in a specific sense defined in Section 3.4.2), assuming that the FHE scheme is semantically secure. This holds both for single-input and multi-input correlation intractability.

3.4.1 Correlation Intractability for Efficiently Searchable Relations

Construction 3.25. *Let* $\mathsf{FHE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval})$ *be any* circular secure *fully homomorphic encryption scheme. We define the following hash family* $\mathcal{H} = \mathcal{H}^{\mathrm{circ}}_{\mathsf{FHE}}$ *associated to* FHE *along with some* circuit size bound $L(\lambda)$, *input length* $n(\lambda)$, *and constant* $\epsilon > 0$*:*

- $\mathcal{H}.\mathsf{Gen}(1^\lambda)$ *calls* $\mathsf{Gen}(1^\lambda)$*, obtaining a pair* $(\mathsf{pk}, \mathsf{sk})$*. It then computes* $\mathsf{ct}_1 = \mathsf{Enc}(\mathsf{pk}, 0^{|\mathsf{sk}|})$, $\mathsf{ct}_2 = \mathsf{Enc}(\mathsf{pk}, 0^L)$*, and outputs* $k = (\mathsf{pk}, \mathsf{ct}_1, \mathsf{ct}_2)$*.*

- $\mathcal{H}.\mathsf{Hash}(k, x)$ *interprets* $k = (\mathsf{pk}, \mathsf{ct}_1, \mathsf{ct}_2)$ *and outputs*

$$y = \mathsf{Eval}(\mathsf{pk}, U_x, \mathsf{ct}_1, \mathsf{ct}_2),$$

where U_x *denotes the universal circuit for evaluation of circuits of size* $L(\lambda)$*, single bit output, and input length* $|x| + |\mathsf{sk}|$*; that is,*

$$U_x(s, C) = C(x, s).$$

We note that the output length of this hash function is some fixed polynomial $\mathsf{poly}(\lambda)$*. By setting the security parameter* λ *appropriately (in relation to* n*), this can result in a hash function with arbitrary polynomial relationship between input and output length.*

Remark 3.26. *One could define the above scheme to use a single ciphertext* ct *(rather than* ct_1 *and* ct_2*) and universal circuit* $U_x(C) = C(x)$*. We take this approach in Section 3.4.2 when we do not use circular security. The advantage of the present formulation, where* C *has an separate input of length* $|\mathsf{sk}|$*, is that it allows for a security proof where the decryption key is an* input *to* C *rather than hardcoded into* C*. This makes it explicit that we only need to assume plain circular security of the FHE in use, rather than general KDM security.*

Theorem 3.27. *Suppose that* FHE *is a* circular secure *fully-homomorphic encryption scheme, let* $n(\lambda) = \lambda^{\Theta(1)}$*, and let* $T = \mathsf{poly}(n, \lambda)$ *be given. Then,* $\mathcal{H} = \mathcal{H}^{\mathrm{circ}}$ *with input length* $n(\lambda)$ *and size parameter* $L = T + \mathsf{poly}(\lambda)$ *is correlation intractable for all relations* R *that are searchable in time* T *(and appropriate input/output lengths).*

In fact, $\mathcal{H}$ *is* somewhere statistically correlation intractable *for this class of relations, such that the function* $\mathsf{StatGen}(1^\lambda, \mathsf{aux})$ *can use any circuit computing the search function* f_R *as its auxiliary input.*

Proof. Let FHE and R be fixed; recall by the definition of T-searchability, there exists a function $f = f_R : \{0,1\}^* \to \{0,1\}^*$ computable in (non-uniform) time T such that if $(x, y) \in R$ then $f(x) = y$.

To show that $\mathcal{H}$ is somewhere statistically correlation intractable for all T-searchable relations R, we define the auxiliary algorithm $\mathcal{H}.\mathsf{StatGen}(1^\lambda, \mathsf{aux})$, which operates as follows:

- Interpret aux as a circuit C of size T.
- Call $\mathsf{Gen}(1^\lambda)$, obtaining a pair $(\mathsf{pk}, \mathsf{sk})$. Then, compute $\mathsf{ct} = \mathsf{Enc}(\mathsf{pk}, (\mathsf{sk}, C'))$, where
$$C'(x, \mathsf{sk}) = 1 \oplus \mathsf{Dec}(\mathsf{sk}, C(x)).$$
- Output $k = (\mathsf{pk}, \mathsf{ct})$.

It now suffices to prove that our augmented hash family $\mathcal{H}$ satisfies key indistinguishability and statistical correlation intractability (for keys generated by $\mathsf{StatGen}$).

- **Key indistinguishability** follows immediately from the circular security of FHE, as the only difference between $\mathcal{H}.\mathsf{Gen}$ and $\mathcal{H}.\mathsf{StatGen}$ is that the former samples $\mathsf{ct} = \mathsf{Enc}(\mathsf{pk}, 0^{|\mathsf{sk}|+L})$ while the latter samples $\mathsf{ct} = \mathsf{Enc}(\mathsf{pk}, (\mathsf{sk}, C'))$.

- **Statistical correlation intractability** holds by the following argument. Let R be any T-searchable relation, let C be any circuit computing the search function f_R, and let $k = (\mathsf{pk}, \mathsf{ct}) \leftarrow \mathsf{StatGen}(1^\lambda, C)$. Then, for any $x \in \{0,1\}^n$ and $y = h(k, x)$, we see that by the correctness of FHE-evaluation,

  $$\mathsf{Dec}(\mathsf{sk}, y) = \mathsf{Dec}(\mathsf{sk}, \mathsf{Eval}(\widetilde{U}_x, \mathsf{ct})) = 1 \oplus \mathsf{Dec}(\mathsf{sk}, C(x)).$$

  Therefore, if $(x, y) \in R$, then $y = f_R(x) = C(x)$ and we obtain the equation $\mathsf{Dec}(\mathsf{sk}, y) = 1 \oplus \mathsf{Dec}(\mathsf{sk}, y)$, a contradiction. Thus, input-output pairs satisfying R (unconditionally) do not exist.

This completes the proof of Theorem 3.27. □

Remark 3.28. *If we assume that* FHE *is* subexponentially secure*, then* $\mathcal{H}^{\mathrm{circ}}$ *is correlation intractable with security* 2^{-m^ϵ} *for some* $\epsilon > 0$*. By Lemma 3.21, this implies that for some* $\epsilon > 0$*,* $\mathcal{H}^{\mathrm{circ}}$ *is correlation intractable for all efficiently sampleable relations with sparsity* $\frac{2^{m^\epsilon}}{2^m}$*.*

Finally, by applying Remark 3.24, we obtain *programmable* CI hash functions for efficiently searchable relations assuming circular-secure FHE.

Corollary 3.29. *Fix functions* $m(\lambda) = n(\lambda)^{\Theta(1)}$*. If circular-secure FHE exists, then for every polynomial function* T*, there exists a* programmable *hash family* $\mathcal{H}$ *that is somewhere statistically correlation intractable for the class of relation ensembles* $R = \{R_\lambda \subseteq \{0,1\}^{n(\lambda)} \times \{0,1\}^{m(\lambda)}\}$ *that are searchable in (non-uniform) time* T*.*

3.4.2 Universal Correlation Intractability from LWE

We now show that a simplified version of Construction 3.25 yields a hash family satisfying interesting notions of *universality* for correlation intractable hash families. We obtain results based on the LWE assumption, either with polynomial or subexponential (that is, 2^{λ^δ}-) security.

Construction 3.30. *Let* $\mathsf{FHE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval})$ *be any (leveled) fully homomorphic encryption scheme. We define the following hash family* $\mathcal{H} = \mathcal{H}^{\text{univ}}_{\mathsf{FHE}}$ *associated to* FHE *along with some* circuit size bound $L(\lambda)$*, input length* $n(\lambda)$*,* plaintext length $m'(\lambda)$*, and constant* $\epsilon > 0$*:*

- $\mathcal{H}.\mathsf{Gen}(1^\lambda)$ *calls* $\mathsf{Gen}(1^{\lambda^\epsilon})$*. It then computes* $\mathsf{ct} = \mathsf{Enc}(\mathsf{pk}, 0^L)$ *and outputs* $k = (\mathsf{pk}, \mathsf{ct})$*.*

- $\mathcal{H}.\mathsf{Hash}(k, x)$ *interprets* $k = (\mathsf{pk}, \mathsf{ct})$ *and outputs*

$$y = \mathsf{Eval}(\mathsf{pk}, U_x, \mathsf{ct}),$$

where U_x *denotes the universal circuit for evaluation of any given circuit* C*, whose size is* $L(\lambda)$ *and whose output length is* $m'(\lambda)$*, on input* x*. That is,* $U_x(C) = C(x)$*.*

We note that the output length of this hash function is not m' *but* $m(\lambda) = \left|\mathsf{Enc}(\mathsf{pk}, 0^{m'(\lambda)})\right| = m' \cdot \mathsf{poly}(\lambda^\epsilon)$*.*

Theorem 3.31. *Suppose that* FHE *is a (leveled) fully-homomorphic encryption scheme, and let* $R = \{R_\lambda \subseteq \{0,1\}^{n(\lambda)} \times \{0,1\}^{m(\lambda) = m'(\lambda)\mathsf{poly}(\lambda^\epsilon)}\}$ *be a relation ensemble that is decidable in polynomial time. Then, if there* exists *a hash family* $\mathcal{H}_R$*, computable by circuits of size at most* L*, that is correlation intractable for all relations on* $\{0,1\}^n \times \{0,1\}^{m'}$ *of the form*

$$R^*_{\mathsf{sk}} = \{(x, z) : \exists y \text{ such that } (x, y) \in R \text{ and } z = \mathsf{Dec}(\mathsf{sk}, y)\},$$

then $\mathcal{H} = \mathcal{H}^{\text{univ}}$ *with parameters* (L, n, m', ϵ) *is correlation intractable for* R*. Moreover, if* FHE *(with security parameter* λ^ϵ*) is secure against* 2^{λ^δ}*-time adversaries, then the same statement holds for all relations* R *decidable in (non-uniform) time* $2^{\lambda^\delta} - \lambda^{\omega(1)}$*.*

Proof. Let FHE and R be fixed. Suppose that some p.p.t. adversary $\mathcal{A}$, given $k \leftarrow \mathcal{H}.\mathsf{Gen}(1^\lambda)$, outputs some $x \in \{0,1\}^n$ such that $(x, h(k,x)) \in R$ with non-negligible probability. Let $\mathcal{H}_R$ denote the correlation intractable hash family hypothesized to exist in the theorem statement.

We first claim that the adversary $\mathcal{A}$ still succeeds with non-negligible probability when given a key $\tilde{k} = (\mathsf{pk}, \mathsf{Enc}(\mathsf{pk}, H_{R,k'}))$, where $H_{R,k'}$ is a circuit computing H_R with randomly sampled key $k' \leftarrow \mathcal{H}_R.\mathsf{Gen}(1^\lambda)$. This follows immediately from the semantic security of FHE, as $\mathcal{A}$'s win condition is decidable in the time required to decide R.

We now describe a p.p.t. adversary $\mathcal{A}'$ that breaks the correlation intractability of $\mathcal{H}_R$:

1. $\mathcal{A}'$ samples FHE parameters $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^{\lambda^\epsilon})$ and declares the relation R^*_{sk} as its challenge.

2. $\mathcal{A}'$ is given a hash key $k' \leftarrow \mathcal{H}_R.\mathsf{Gen}(1^\lambda)$.

3. $\mathcal{A}'$ computes $\mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{pk}, H_{R,k'})$, and runs $\mathcal{A}'((\mathsf{pk}, \mathsf{ct}))$.

4. $\mathcal{A}'$ obtains an input $x \in \{0,1\}^n$ and returns x.

By construction, whenever $\mathcal{A}((\mathsf{pk}, \mathsf{ct}))$ breaks the R-correlation intractability of $\mathcal{H}$, we have that $\mathcal{A}'$ breaks the R^*_{sk}-correlation intractability of $\mathcal{H}_R$. To see this, note that if $y = h((\mathsf{pk}, \mathsf{ct}), x)$ in the above experiment, then

$$\mathsf{Dec}(\mathsf{sk}, y) = \mathsf{Dec}(\mathsf{sk}, \mathsf{Eval}(\mathsf{pk}, U_x, \mathsf{ct})) = h_R(k', x).$$

Thus, if $(x, y) \in R$, then $(x, h_R(k', x)) \in R^*_{\mathsf{sk}}$. This completes the proof of Theorem 3.31. □

As an immediate corollary of Theorem 3.31, we conclude that $\mathcal{H}^{\text{univ}}$ is weakly *universal* for correlation intractable hash families (for sufficiently sparse relations), in the following sense.

Corollary 3.32. *Let $\gamma < 1$ and $\delta < 1$ be arbitrary, and let $m'(\lambda) = \lambda^{\Omega(1)}$ grow at least polynomially with λ. Set*

$$\epsilon = \Omega\left(\frac{\delta}{1-\delta}\log_\lambda(m')\right)$$

and

$$\beta = \frac{\gamma}{1-\delta}.$$

Finally, suppose that there exists a hash family (computable by a size L circuit) $\mathcal{H}_\beta$ that is correlation intractable for all relations *on $\{0,1\}^n \times \{0,1\}^{m'}$ with sparsity $\frac{2^{m'^\beta}}{2^{m'}}$. Then, assuming the security of FHE, $\mathcal{H}^{\mathrm{univ}}$ implemented with parameters (L, n, m', ϵ) is correlation intractable for all* efficiently decidable *relations on $\{0,1\}^n \times \{0,1\}^m$ of sparsity $\frac{2^{m'^\beta}}{2^m} = \frac{2^{m^\gamma}}{2^m}$. The same is true for all (sufficiently sparse) subexponentially decidable relations on $\{0,1\}^n \times \{0,1\}^m$ if FHE is assumed to be subexponentially secure.*

Finally, we note that Theorem 3.27 – our construction of CI for efficiently searchable relations from circular secure FHE – can be thought of as a twist on the proof of Theorem 3.31. The difference between the two proofs is that in Theorem 3.27, we use the circular security of FHE to reduce from the security of $\mathcal{H}^{\mathrm{circ}}$ for f to the existence of a hash family $\mathcal{H}_{f,\mathsf{sk}}$ that is correlation intractable for *single* relation that depends on f and sk, the FHE secret key. Moreover, again due to the circular security assumption, we can use sk in the construction of $\mathcal{H}_{f,\mathsf{sk}}$. This allows for an unconditional construction of the "inner hash function" that we have to assume exists in Theorem 3.31.

3.4.3 Multi-Input Correlation Intractability

Our universality results even extend to the notion of *multi-input* correlation intractability, about which very little is known.[9]

[9] The only known instantiations of multi-input correlation intractable hash families focus on the special case where the relation depends only on the output [Zha16, HL18] or rely on indistinguishability obfuscation [HL18].

Definition 3.33 (Multi-Input Correlation Intractability)**.** *Let ℓ (possibly depending on λ) denote an* arity*. For a given relation ensemble $R = \{R_\lambda \subseteq (\{0,1\}^{n(\lambda)})^{\ell(\lambda)} \times (\{0,1\}^{m(\lambda)})^{\ell(\lambda)}\}$, a hash family $\{h_\lambda : \{0,1\}^{s(\lambda)} \times \{0,1\}^{n(\lambda)} \to \{0,1\}^{m(\lambda)}\}$ is said to be* R-correlation intractable *if for every polynomial-size $\mathcal{A} = \{\mathcal{A}_\lambda\}$,*

$$\Pr_{\substack{k \leftarrow \mathcal{H}.\mathsf{Gen}(1^\lambda) \\ \mathbf{x}=(x_1,\ldots,x_\ell) \leftarrow \mathcal{A}(k)}} \Big[\Big(x_1, \ldots, x_\ell, h(k, x_1), \ldots, h(k, x_\ell)\Big) \in R\Big] = \mathrm{negl}(\lambda)$$

If $\mathcal{R}$ is a collection of relation ensembles, then $\mathcal{H}$ is said to be uniformly $\mathcal{R}$-correlation intractable *if for every polynomial-size $\mathcal{A}$, there exists a function $\nu(\lambda) = \mathrm{negl}(\lambda)$ such that for every $R \in \mathcal{R}$,*

$$\Pr_{\substack{k \leftarrow \mathcal{H}.\mathsf{Gen}(1^\lambda) \\ \mathbf{x}=(x_1,\ldots,x_\ell) \leftarrow \mathcal{A}(k)}} \Big(x_1, \ldots, x_\ell, h(k, x_1), \ldots, h(k, x_\ell)\Big) \leq \nu(\lambda).$$

Analogously to Theorem 3.31, we observe that Construction 3.30 satisfies interesting notions of universality for *multi-input* correlation intractable hash families.

Theorem 3.34. *Suppose that* FHE *is a (leveled) fully-homomorphic encryption scheme, and let $R = \{R_\lambda \subseteq (\{0,1\}^{n(\lambda)})^{\ell(\lambda)} \times (\{0,1\}^{m(\lambda) = m'(\lambda)\mathsf{poly}(\lambda^\epsilon)})^{\ell(\lambda)}\}$ be a relation ensemble that is decidable in polynomial time. Then, if there* exists *a hash family $\mathcal{H}_R$, computable by circuits of size at most L, that is correlation intractable for all relations on $(\{0,1\}^n)^\ell \times (\{0,1\}^{m'})^\ell$ of the form*

$$R^*_{\mathsf{sk}} = \{(\mathbf{x}, \mathbf{z}) : \exists \mathbf{y} \textit{ such that } (\mathbf{x}, \mathbf{y}) \in R \textit{ and } \mathbf{z} = \mathsf{Dec}(\mathsf{sk}, \mathbf{y})\},$$

then $\mathcal{H} = \mathcal{H}^{\mathrm{univ}}$ with parameters (L, n, m', ϵ) is correlation intractable for R. Moreover, if FHE *(with security parameter λ^ϵ) is secure against 2^{λ^δ}-time adversaries, then the same statement holds for all relations R decidable in (non-uniform) time $2^{\lambda^\delta} - \lambda^{\omega(1)}$.*

Proof. This is largely identical to the proof of Theorem 3.31, but we include a proof for completeness.

Let FHE and R be fixed. Suppose that some p.p.t. adversary $\mathcal{A}$, given $k \leftarrow$

$\mathcal{H}.\mathsf{Gen}(1^\lambda)$, outputs some $\mathbf{x} \in (\{0,1\}^n)^\ell$ such that $(\mathbf{x}, h(k, x_1), \ldots, h(k, x_\ell)) \in R$ with non-negligible probability. Let $\mathcal{H}_R$ denote the correlation intractable hash family hypothesized to exist in the theorem statement.

We first claim that the adversary $\mathcal{A}$ still succeeds with non-negligible probability when given a key $\tilde{k} = (\mathsf{pk}, \mathsf{Enc}(\mathsf{pk}, H_{R,k'}))$, where $H_{R,k'}$ is a circuit computing H_R with randomly sampled key $k' \leftarrow \mathcal{H}_R.\mathsf{Gen}(1^\lambda)$. This follows immediately from the semantic security of FHE, as $\mathcal{A}$'s win condition is decidable in the time required to decide R.

We now describe a p.p.t. adversary $\mathcal{A}'$ that breaks the correlation intractability of $\mathcal{H}_R$:

1. $\mathcal{A}'$ samples FHE parameters $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^{\lambda^\epsilon})$ and declares the relation R^*_{sk} as its challenge.

2. $\mathcal{A}'$ is given a hash key $k' \leftarrow \mathcal{H}_R.\mathsf{Gen}(1^\lambda)$.

3. $\mathcal{A}'$ computes $\mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{pk}, H_{R,k'})$, and runs $\mathcal{A}'(\mathsf{pk}, \mathsf{ct})$.

4. $\mathcal{A}'$ obtains an input $\mathbf{x} \in (\{0,1\}^n)^\ell$ and returns $\mathbf{x}$.

By construction, whenever $\mathcal{A}(\mathsf{pk}, \mathsf{ct})$ breaks the R-correlation intractability of $\mathcal{H}$, we have that $\mathcal{A}'$ breaks the R^*_{sk}-correlation intractability of $\mathcal{H}_R$. To see this, note that if $y_i = h((\mathsf{pk}, \mathsf{ct}), x_i)$ in the above experiment, then

$$\mathsf{Dec}(\mathsf{sk}, y_i) = \mathsf{Dec}(\mathsf{sk}, \mathsf{Eval}(\mathsf{pk}, U_{x_i}, \mathsf{ct})) = h_R(k', x_i).$$

Thus, if $(\mathbf{x}, \mathbf{y}) \in R$, then $(\mathbf{x}, h_R(k', x_1), \ldots, h_R(k', x_\ell)) \in R^*_{\mathsf{sk}}$. This completes the proof of Theorem 3.34. $\square$

It follows, as a corollary of Theorem 3.34, that if there *exists* a hash family that is correlation intractable for all (sufficiently sparse) multi-input relations, then for an appropriate parameter setting, $\mathcal{H}^{\mathrm{univ}}$ is correlation intractable for all (efficiently decidable, sufficiently sparse) multi-input relations.

Although little is known about the existence of *general* multi-input correlation intractable hash families, the security reduction in Theorem 3.34 maps output relations (i.e. relations $R(\mathbf{x}, \mathbf{y})$ that depend only on $\mathbf{y}$) to output relations, so we obtain a concrete hash function that combines a family of candidates from [HL18].[10] As an example, we obtain the following corollary.

Corollary 3.35. *Assume the hardness of* LWE. *In addition, assume that there* exists *family of (symmetric, injective) k-one way product functions (OWPFs) with security* $2^{-kn^{\beta}} \cdot \mathrm{negl}(n)$ *for some $\beta < 1$.*[11]

Then for arbitrary $\gamma < \beta$, the hash family $\mathcal{H}^{\mathrm{univ}}$ (for appropriate parameter settings) using an LWE*-based (leveled) FHE scheme is correlation intractable for efficiently decidable* output *relations of sparsity* $2^{kn^{\gamma} - kn}$.

Note that we only need to assume that the [HL18] OWPFs *exist*; we do not need an explicit description of one in the construction of $\mathcal{H}^{\mathrm{univ}}$. This is similar to the obfuscation-based result of [HL18], but Corollary 3.35 replaces indistinguishability obfuscation with LWE. However, our result only applies in the regime of fairly low ($2^{kn^{\gamma} - kn}$) sparsity.

3.5 Non-Interactive Zero Knowledge Arguments

In this section, we apply Theorem 3.27 to obtain Theorem 3.2, that is, NIZK arguments for **NP** assuming circular secure FHE. This closely follows the framework of [HL18, CCH+18] for obtaining NIZK arguments from weak forms of correlation intractability, but we apply the framework to the 3-message [FLS90] protocol for graph Hamiltonicity. This allows us to rely on correlation intractable hash functions for efficiently searchable relations (as constructed in Theorem 3.27) as opposed to efficiently samplable relations (as defined in [HL18, CCH+18]).

We augment our basic NIZK argument system in two different ways:

[10] The [Zha16] construction does not give a hash family that is correlation intractable for all output relations – only efficiently decidable relations – so we cannot use it as is.

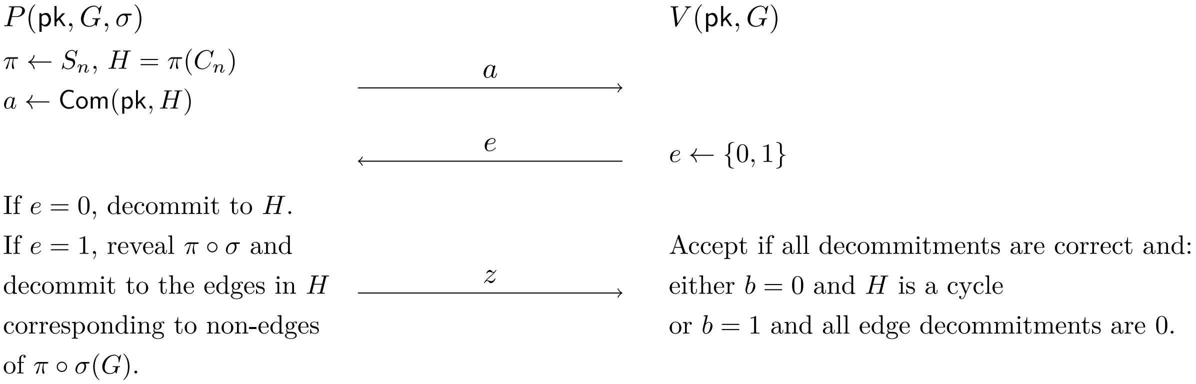
[11] We refer the reader to [HL18] for a discussion of OWPFs.

- By using our *somewhere statistically correlation intractable* hash functions (Definition 3.15), we show that our NIZK argument system has a *statistically sound mode*, yielding NIZK *proofs* in the common reference string model.
- By using a lossy public key encryption scheme with uniformly lossy public keys (Definition 3.7), we show that our NIZK argument system has a *statistical zero knowledge mode* in which the CRS is uniformly random, yielding NISZK arguments in the common *random* string model.

We begin by recalling the 3-message [FLS90] protocol.

3.5.1 The [FLS90] Protocol

We construct NIZK arguments by applying the Fiat-Shamir transform to a variant of the 3-message [FLS90] proof system for graph Hamiltonicity. Recall that the Hamiltonicity language L_{Ham} consists of all graphs G with a Hamitonian cycle, and the standard **NP**-relation for L_{Ham} uses a permutation σ as a witness exhibiting a Hamiltonian cycle $\sigma^{-1}(C_n)$ in G. We describe the 3-message protocol $\Pi = \Pi_{\text{FLS}}$ in Fig. 3-2. For the purpose of obtaining adaptive zero knowledge, we recall that this protocol is "delayed input," namely the prover need to know the graph G and the cycle σ only for computing the third message. In our proof below, we will make use of the following facts.



$P(\mathsf{pk}, G, \sigma)$

$V(\mathsf{pk}, G)$

$\pi \leftarrow S_n$, $H = \pi(C_n)$

$a \leftarrow \mathsf{Com}(\mathsf{pk}, H)$

$\xrightarrow{a}$

$\xleftarrow{e}$

$e \leftarrow \{0,1\}$

If $e = 0$, decommit to H.
If $e = 1$, reveal $\pi \circ \sigma$ and decommit to the edges in H corresponding to non-edges of $\pi \circ \sigma(G)$.

$\xrightarrow{z}$

Accept if all decommitments are correct and:
either $b = 0$ and H is a cycle
or $b = 1$ and all edge decommitments are 0.

Figure 3-2: The Zero Knowledge Proof System Π_{FLS} for Graph Hamiltonicity.

Fact 3.36. *For any computationally hiding commitment scheme* Com *(potentially in the CRS model),* Π *is honest-verifier zero knowledge. If* Com *is a statistically hiding commitment scheme, then* Π *is honest-verifier* statistical *zero knowledge.*

Finally, Π *is "honest-verifier adaptive zero knowledge", meaning that* Π *remains honest-verifier zero knowledge when the adversary is allowed to choose* (x, w) *as an (arbitrary) efficient function of the transcript* $(\mathsf{crs}, \mathbf{a}, \mathbf{e})$ *up to the second message.*

We emphasize that our variant of the [FLS90] protocol explicitly allows for the commitment scheme Com to rely on a public commitment key pk.

3.5.2 Our NIZK Protocol

We start by defining three different modes for our protocol and use them to help prove security. We use the following tools in our construction.

- A hash family $\mathcal{H}$ satisfying two properties:

  - Correlation intractability for all (subexponentially sparse) relations that are (non-uniformly) searchable in a fixed polynomial time T.
  - Programmability, as in Definition 3.22.

  In Construction 3.38, we will assume that $\mathcal{H}$ is somewhere statistically correlation intractable for the above class of relations, and that $\mathcal{H}.\mathsf{StatGen}(1^\lambda, C)$ can use any circuit C computing the search function f_R as auxiliary input. In Construction 3.39, we will assume that $\mathcal{H}$ has pseudorandom keys, and that the modified hash family $\mathcal{H}'$ using a uniformly random key is also programmable.[12]

- A public key encryption scheme $\mathsf{PKE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$. In Construction 3.39, we will assume that PKE is lossy (Definition 3.7) with uniformly random lossy public keys.

[12] The generic transformation (Construction 3.23) from correlation intractable hash families to programmable correlation intractable hash families guarantees this property.

Construction 3.37 (Basic NIZK)**.** *Let $\Pi = \Pi_{\text{FLS}}$ be the [FLS90] protocol from Section 3.5.1, in which we instantiate the commitment scheme* Com *in the CRS model using* PKE *in the natural way:* $\mathsf{Com}(\mathsf{pk}, b; \rho) = \mathsf{Enc}(\mathsf{pk}, b; \rho)$*. We apply the Fiat-Shamir transform, using $\mathcal{H}$, to the protocol Π^λ; that is, the protocol Π repeated λ times in parallel. We call the resulting protocol $\widetilde{\Pi}$, which is formally defined as follows.*

- *Common reference string: a* PKE*-public key* pk *along with a hash key $k \leftarrow \mathcal{H}.\mathsf{Gen}(1^\lambda)$.*

- *Prover message: given an instance x, witness w, and common reference string* $\mathsf{crs} = (\mathsf{pk}, k)$*, the prover computes $\mathbf{a} \leftarrow \Pi^\lambda.P(\mathsf{crs}, x, w)$, $\mathbf{e} = h(k, \mathbf{a})$, $\mathbf{z} = \Pi^\lambda.P(\mathsf{crs}, x, w, \mathbf{a}, \mathbf{e})$, and outputs $(\mathbf{a}, \mathbf{e}, \mathbf{z})$.*

- *The verifier accepts a transcript $(\mathsf{crs}, x, \mathbf{a}, \mathbf{e}, \mathbf{z})$ if $\mathbf{e} = h(k, \mathbf{a})$ and $\Pi^\lambda.V(\mathsf{crs}, x, \mathbf{a}, \mathbf{e}, \mathbf{z}) = 1$.*

Our two modified constructions change only the common reference string distribution.

Construction 3.38 (Statistically Sound Mode)**.** *Let $\Pi = \Pi_{\text{FLS}}$ be the [FLS90] protocol from Section 3.5.1, in which we instantiate the commitment scheme* Com *in the CRS model using* PKE*. The statistically sound mode of our protocol $\widetilde{\Pi}_{\text{Sound}}$ is then defined as follows.*

- *Common reference string: a* PKE*-public key* pk *along with a fake hash key $k \leftarrow \mathcal{H}.\mathsf{StatGen}(1^\lambda, C_{\mathsf{sk}})$. Here,* sk *is the* PKE*-secret key associated to* pk*, and C_{sk} is a (poly-size) circuit computing the function $f_{\mathsf{sk}}(\mathbf{a}) = \mathbf{e}$ such that for every $i \in [\lambda]$, $e_i = 0$ if and only if $\mathsf{Dec}(\mathsf{sk}, a_i)$ is a cycle.*

- *Prover message: given an instance x, witness w, and common reference string* $\mathsf{crs} = (\mathsf{pk}, k)$*, the prover computes $\mathbf{a} \leftarrow \Pi^\lambda.P(\mathsf{crs}, x, w)$, $\mathbf{e} = h(k, \mathbf{a})$, $\mathbf{z} = \Pi^\lambda.P(\mathsf{crs}, x, w, \mathbf{a}, \mathbf{e})$, and outputs $(\mathbf{a}, \mathbf{e}, \mathbf{z})$.*

- *The verifier accepts a transcript $(\mathsf{crs}, x, \mathbf{a}, \mathbf{e}, \mathbf{z})$ if $\mathbf{e} = h(k, \mathbf{a})$ and $\Pi^\lambda.V(\mathsf{crs}, x, \mathbf{a}, \mathbf{e}, \mathbf{z}) = 1$.*

Construction 3.39 (Statistical Zero Knowledge Mode)**.** *Let* $\Pi = \Pi_{\mathrm{FLS}}$ *be the [FLS90] protocol from Section 3.5.1, in which we instantiate the commitment scheme* Com *in the CRS model using* PKE*. The statistical zero knowledge mode of our protocol* $\widetilde{\Pi}_{\mathrm{ZK}}$ *is then defined as follows.*

- *Common reference string: a (uniformly random)* PKE*-lossy public key* $\mathsf{pk} \leftarrow \mathsf{FakeGen}(1^\lambda)$ *along with a uniformly random hash key* k*.*

- *Prover message: given an instance* x*, witness* w*, and common reference string* $\mathsf{crs} = (\mathsf{pk}, k)$*, the prover computes* $\mathbf{a} \leftarrow \Pi^\lambda.P(\mathsf{crs}, x, w)$*,* $\mathbf{e} = h(k, \mathbf{a})$*,* $\mathbf{z} = \Pi^\lambda.P(\mathsf{crs}, x, w, \mathbf{a}, \mathbf{e})$*, and outputs* $(\mathbf{a}, \mathbf{e}, \mathbf{z})$*.*

- *The verifier accepts a transcript* $(\mathsf{crs}, x, \mathbf{a}, \mathbf{e}, \mathbf{z})$ *if* $\mathbf{e} = h(k, \mathbf{a})$ *and* $\Pi^\lambda.V(\mathsf{crs}, x, \mathbf{a}, \mathbf{e}, \mathbf{z}) = 1$*.*

Theorem 3.40. $\widetilde{\Pi}$ *is a NIZK argument system for* **NP** *in the common reference string model satisfying both adaptive soundness and adaptive zero knowledge. Moreover,* $\widetilde{\Pi}_{\mathrm{Sound}}$ *is a NIZK* proof *system for* **NP** *in the common reference string model satisfying adaptive zero knowledge, and* $\widetilde{\Pi}_{\mathrm{ZK}}$ *is a* non-adaptively *sound* NISZK *argument system for* **NP** *in the common* random *string model.*

We now proceed to prove Theorem 3.40 by proving a sequence of lemmas about our construction.

Lemma 3.41. $\widetilde{\Pi}$ *is (adaptively) sound.*

Proof. To see this, we argue that $\widetilde{\Pi}$ is adaptively sound if $\mathcal{H}$ is correlation intractable for all relations of the form

$$R_{\mathsf{sk}} = \{(\mathbf{a}, \mathbf{e}) : \mathbf{e} = f_{\mathsf{sk}}(\mathbf{a})\}.$$

This holds because if $\mathsf{Dec}(\mathsf{sk}, a_i)$ is not a cycle and $e_i = 0$, then there is no input graph x and third message z_i such that (x, a_i, e_i, z_i) is an accepting transcript in the original protocol Π. Similarly, if $\mathsf{Dec}(\mathsf{sk}, a_i)$ is a cycle and $e_i = 1$, then there is no input graph x *that is not Hamiltonian* (i.e. there is no false statement x) and third message z_i such that (x, a_i, e_i, z_i) is an accepting transcript in Π. These two facts make use of the perfect decryption correctness of PKE and the standard (adaptive) soundness analysis of Π.

From this analysis, we conclude that any adversary $\mathcal{A}$ breaking the (adaptive) soundness of $\widetilde{\Pi}$ breaks the correlation intractability of $\mathcal{H}$ with respect to some sk (indeed, a random sk sampled according to $\mathsf{PKE.Gen}$ suffices).

Finally, we note that for every secret key sk, R_{sk} is an efficiently searchable relation: indeed, the function f_{sk} is efficiently computable given sk. Thus, since we assumed $\mathcal{H}$ is correlation intractable for all efficiently searchable relations, we conclude that $\widetilde{\Pi}$ is adaptively sound. □

Lemma 3.42. $\widetilde{\Pi}$ *is (adaptive) zero knowledge.*

Proof. The proof that $\widetilde{\Pi}$ is zero knowledge is almost identical to the proof of zero knowledge in ([CCH+18] Theorem 7.7). For completeness, we describe a simulator for $\widetilde{\Pi}$:

- **Input:** a graph x.
- Sample $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{PKE.Gen}(1^\lambda)$.
- Call the honest verifier simulator $\Pi^t.\mathsf{HVSim}(x, \mathsf{pk})$ associated to the parallel repeated protocol Π^t, producing $(\mathbf{a}, \mathbf{e}, \mathbf{z})$.
- Sample a hash key $\widetilde{k}$ from the conditional distribution $k \leftarrow \mathcal{H}.\mathsf{Gen}(1^\lambda) \mid h(k, \mathbf{a}) = \mathbf{e}$.
- Output $(\mathsf{pk}, \widetilde{k}, \mathbf{a}, \mathbf{e}, \mathbf{z})$.

Zero knowledge then follows from Fact 3.36 and the programmability of $\mathcal{H}$ (by a standard hybrid argument).

To see that $\widetilde{\Pi}$ is *adaptive* zero knowledge, we use the fact that Π is honest-verifier adaptive zero knowledge (and use this two-part simulator in place of HVSim above).

We refer the reader to [CCRR18] (Proposition 7.6) for more details on obtaining adaptive zero knowledge using Fiat-Shamir. □

This completes the proof that $\widetilde{\Pi}$ is an adaptively sound NIZK argument system in the common reference string model.

Lemma 3.43. $\widetilde{\Pi}_{\text{Sound}}$ *is statistically sound.*

Proof. Let $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^\lambda)$ and $k \leftarrow \mathcal{H}.\mathsf{StatGen}(1^\lambda, C_{\mathsf{sk}})$ as in Construction 3.38. Then, for any first message $\mathbf{a}$ for the protocol Π^λ, if $\mathsf{Dec}(\mathsf{sk}, a_i)$ is not a cycle and $e_i = 0$, then there is no input graph G and third message z_i such that (G, a_i, e_i, z_i) is an accepting transcript in the original protocol Π. Similarly, if $\mathsf{Dec}(\mathsf{sk}, a_i)$ is a cycle and $e_i = 1$, then there is no input graph G *that is not Hamiltonian* (i.e. there is no false statement G) and third message z_i such that (G, a_i, e_i, z_i) is an accepting transcript in Π. These two facts make use of the perfect decryption correctness of PKE and the standard (adaptive) soundness analysis of Π.

This tells us that any accepting transcript $(\mathsf{pk}, G, \mathbf{a}, \mathbf{e}, \mathbf{z})$ for $\widetilde{\Pi}_{\text{Sound}}$ must satisfy $\mathbf{e} = f_{\mathsf{sk}}(\mathbf{a})$. However, by the statistical correlation intractability of $\mathcal{H}$, we know that there does not exist an input $\mathbf{a}$ such that $h(k, \mathbf{a}) = f_{\mathsf{sk}}(\mathbf{a})$, so we conclude that $\widetilde{\Pi}_{\text{Sound}}$ is statistically sound. □

Lemma 3.44. *For any p.p.t. verifier* $V^*(\mathsf{crs})$ *outputting a triple* (x, w, aux)*, honestly generated transcripts in* $\widetilde{\Pi}$ *and* $\widetilde{\Pi}_{\text{Sound}}$ *are computationally indistinguishable (even given* (x, w, aux)*).*

Proof. This follows immediately from the key indistinguishability of $\mathcal{H}$. □

Combining Lemma 3.43, Lemma 3.44, and Lemma 3.42, we conclude that Construction 3.38 is a NIZK proof system for **NP** in the common reference string model satisfying adaptive zero knowledge.

Lemma 3.45. $\widetilde{\Pi}_{\text{ZK}}$ *is statistical zero knowledge.*

Proof. We define a simulator for $\widetilde{\Pi}_{\text{ZK}}$ as follows.

- **Input:** a graph x.

- Sample a uniformly random public key $\widetilde{\mathsf{pk}}$.

- Call the statistical honest verifier simulator $\Pi^\lambda.\mathsf{HVSim}(x, \mathsf{pk})$ associated to the parallel repeated protocol Π^λ, producing $(\mathbf{a}, \mathbf{e}, \mathbf{z})$.

- Sample a hash key $\widetilde{k}$ from the conditional distribution $k \leftarrow \mathcal{H}'.\mathsf{Gen}(1^\lambda) \mid h(k, \mathbf{a}) = \mathbf{e}$, where $\mathcal{H}'$ is the modified hash family using uniformly random hash keys.

- Output $(\widetilde{\mathsf{pk}}, \widetilde{k}, \mathbf{a}, \mathbf{e}, \mathbf{z})$.

Zero knowledge then follows directly from the lossiness of PKE (which implies that the resulting commitment scheme is statistically hiding), Fact 3.36, and the programmability of $\mathcal{H}'$ (by a standard hybrid argument). □

Lemma 3.46. *The common reference strings in $\widetilde{\Pi}$ and $\widetilde{\Pi}_{\mathrm{ZK}}$ are computationally indistinguishable.*

Proof. This follows immediately from the key indistinguishability of PKE (between real and lossy keys) and the pseudorandomness of the $\mathcal{H}$-keys. □

Combining Lemma 3.45, Lemma 3.46 and Lemma 3.41, we conclude that $\widetilde{\Pi}_{\mathrm{ZK}}$ is a non-adaptively sound[13] NISZK argument system for **NP** in the common random string model. This completes the proof of Theorem 3.40.

3.5.3 Obtaining Theorem 3.3, Theorem 3.4, and LWE-based Instantiation

Recall the statements of Theorem 3.3 and Theorem 3.4, our main results on obtaining NIZK arguments.

[13] We only obtain non-adaptive soundness because switching modes (i.e., invoking computational indinguishability between two CRS distributions) is only guaranteed to preserve non-adaptive soundness. This is because the adversary's win condition in the adaptive soundness security game – producing (x, π) such that $x \notin L$ and π is an accepting proof – is not efficiently checkable; it may not be possible to efficiently verify that $x \notin L$. Relatedly, [Pas13] proves a black-box impossibility result for constructing adaptively sound NISZK arguments from falsifiable assumptions.

Theorem 3.47. *Suppose that circular-secure fully homomorphic encryption exists. Then, there exist NIZK proofs for* **NP** *in the common reference string model.*

Theorem 3.48. *Suppose that there exists a circular-secure fully homomorphic encryption scheme with pseudorandom ciphertexts and public keys. Furthermore, suppose that there exists a lossy public key encryption scheme [KN08, PVW08, BHY09] with uniformly random lossy public keys. Then, there exist (non-adaptively sound) NISZK arguments for* **NP** *in the common random string model.*

We obtain these results by a direct combination of Theorem 3.27 and Theorem 3.40. Theorem 3.40 states that (1) the desired NIZK proofs follow from the existence of somewhere statistically correlation intractable hash functions for (subexponentially sparse) efficiently searchable relations, and (2) under the lossy PKE assumption, the desired NIZK arguments follow from the existence of correlation intractable hash functions for (subexponentially sparse) efficiently searchable relations (with pseudorandom hash keys). Theorem 3.27 states that assuming circular secure FHE (with pseudorandom ciphertexts and public keys), such hash families exist.

Moreover, we note that both of the generic primitives in Theorem 3.4 can be instantiated from (circular secure) LWE. Namely, under plain LWE, Regev encryption [Reg05] is a lossy PKE scheme and has uniformly random lossy public keys, and under various circular security assumptions on LWE [BV11, BGV12, Bra12, GSW13, BV14], there exist circular secure FHE schemes.

3.6 Fiat-Shamir for (Instance-Dependent) Trapdoor Σ-protocols

In this section, we formalize our notions of "trapdoor Σ-protocols" and "instance-dependent trapdoor Σ-protocols," and we prove that our hash family from Theorem 3.27 suffices to instantiate the Fiat-Shamir heuristic for such protocols. Examples of (instance-dependent) trapdoor Σ-protocols include variants of the [Blu86] and [FLS90] protocols for graph Hamiltonicity (in which the commitment scheme is instantiated using public-key encryption as in Section 3.5) as well as the *unmodified* [GMR85] protocol for quadratic residuosity. By the connection between Fiat-Shamir and (malicious verifier) zero knowledge [DNRS99], we conclude that these protocols cannot be malicious verifier zero knowledge, assuming the existence of circular-secure FHE and the hardness of deciding the underlying languages. This partially resolves open questions due to [DNRS99, BLV03].

3.6.1 Instance-Dependent Trapdoor Σ-Protocols

We provide the following definition of a Σ-protocol, which suffices for our purposes. We do not require any extractability ("proof of knowledge") property.

Definition 3.49 (Σ-Protocol)**.** *We say that a three-message honest-verifier zero-knowledge proof system* $\Pi = (\mathsf{Gen}, P, V)$ *in the common reference string model is a* Σ-$\mathsf{protocol}$ *if for every common reference string* crs*, every instance* $x \notin L$*, and every first message* $\mathbf{a}$*, there is* at most one challenge $\mathbf{e} := f(\mathsf{crs}, x, \mathbf{a})$ *such that* $(\mathsf{crs}, x, \mathbf{a}, \mathbf{e}, \mathbf{z})$ *is an accepting transcript* for any choice of third message $\mathbf{z}$*.*

We informally call f *the "bad-challenge function" associated to* Π*, and note that* f *may not be efficiently computable.*

We now define a trapdoor Σ-protocol to be, roughly speaking, a Σ-protocol that has a trapdoor making the bad-challenge function f efficiently computable.

Definition 3.50 (Trapdoor Σ-Protocol)**.** *We say that a* Σ*-protocol* $\Pi = (\mathsf{Gen}, P, V)$ *with bad-challenge function* f *is a* $\mathsf{trapdoor}$ Σ-$\mathsf{protocol}$ *if there are p.p.t. algorithms* $\mathsf{TrapGen}, \mathsf{BadChallenge}$ *with the following syntax.*

- $\mathsf{TrapGen}(1^\lambda)$ *takes as input the security parameter. It outputs a common reference string* crs *along with a trapdoor* τ*.*
- $\mathsf{BadChallenge}(\tau, \mathsf{crs}, x, \mathbf{a})$ *takes as input a trapdoor* τ*, common reference string* crs*, instance* x*, and first message* $\mathbf{a}$*. It outputs a challenge* $\mathbf{e}$*.*

We additionally require the following properties.

- ***CRS Indistinguishability:*** *An honestly generated common reference string* crs *is computationally indistinguishable from a common reference string output by* $\mathsf{TrapGen}(1^\lambda)$.
- ***Correctness:*** *for every instance* $x \notin L$ *and for all* $(\mathsf{crs}, \tau) \leftarrow \mathsf{TrapGen}(1^\lambda)$, *we have that* $\mathsf{BadChallenge}(\tau, \mathsf{crs}, x, \mathbf{a}) = f(\mathsf{crs}, x, \mathbf{a})$.

While this definition is enough to capture our modification to the [FLS90] protocol, it is necessarily limited to Σ-protocols that have a common reference string. To capture the *unmodified* [GMR85] protocol, we generalize our definitition so that the trapdoor τ can depend on the instance x.

Definition 3.51 (Instance-Dependent Trapdoor Σ-Protocol)**.** *We say that a* Σ*-protocol* $\Pi = (\mathsf{Gen}, P, V)$ *with bad-challenge function* f *is an* instance-dependent trapdoor Σ-protocol *if there are p.p.t. algorithms* $\mathsf{TrapGen}, \mathsf{BadChallenge}$ *with the following syntax.*

- $\mathsf{TrapGen}(1^\lambda, x, \mathsf{aux})$ *takes as input the security parameter, an instance* x, *and an auxiliary input* aux. *It outputs a common reference string* crs *along with a trapdoor* τ.
- $\mathsf{BadChallenge}(\tau, \mathsf{crs}, x, \mathbf{a})$ *takes as input a trapdoor* τ, *common reference string* crs, *instance* x, *and first message* $\mathbf{a}$. *It outputs a challenge* $\mathbf{e}$.

We additionally require the following properties.

- ***CRS Indistinguishability:*** *For any* (x, aux), *an honestly generated common reference string* crs *is computationally indistinguishable from a common reference string output by* $\mathsf{TrapGen}(1^\lambda, x, \mathsf{aux})$.
- ***Correctness:*** *for every instance* $x \notin L$, *there exists an auxiliary input* aux *such that for all* $(\mathsf{crs}, \tau) \leftarrow \mathsf{TrapGen}(1^\lambda, x, \mathsf{aux})$, *we have that* $\mathsf{BadChallenge}(\tau, \mathsf{crs}, x, \mathbf{a}) = f(\mathsf{crs}, x, \mathbf{a})$.

Given this definition, we can now state our result on Fiat-Shamir for instance-dependent trapdoor Σ-protocols.

Theorem 3.52. *Suppose that $\mathcal{H}$ is a hash family that is correlation-intractable for all subexponentially sparse relations that are searchable in time T. Moreover, suppose that $\Pi = (\mathsf{Gen}, P, V, \mathsf{TrapGen}, \mathsf{BadChallenge})$ is an instance-dependent trapdoor Σ-protocol with $2^{-\lambda^{\epsilon}}$ soundness for some $\epsilon > 0$, such that $\mathsf{BadChallenge}(\tau, \mathsf{crs}, x, \mathbf{a})$ is computable in time T. Then, $\mathcal{H}$ soundly instantiates the Fiat-Shamir heuristic for Π.*

Proof. Let $\widetilde{\Pi}$ denote the one-message protocol resulting from applying the Fiat-Shamir transform, using $\mathcal{H}$, to Π. Explicitly, $\widetilde{\Pi}$ is defined as follows.

- The **common reference string** consists of a common reference string crs_{Π} associated to Π, along with a hash key $k \leftarrow \mathcal{H}.\mathsf{Gen}(1^{\lambda})$.

- **Prover message:** a triple $(\mathbf{a}, \mathbf{e}, \mathbf{z})$, where $\mathbf{a}$ is computed by $\Pi.P(\mathsf{crs}_{\Pi}, x, w)$, $\mathbf{e} = h(k, \mathbf{a})$, and $\mathbf{z}$ is computed by $\Pi.P(\mathsf{crs}_{\Pi}, x, w, \mathbf{a}, \mathbf{e})$.

- The verifier accepts a transcript $(\mathsf{crs}_{\Pi}, k, x, \mathbf{a}, \mathbf{e}, \mathbf{z})$ if $\Pi.V(\mathsf{crs}_{\Pi}, \mathbf{a}, \mathbf{e}, \mathbf{z}) = 1$ and $\mathbf{e} = h(k, \mathbf{a})$.

By construction, and by the definition of a Σ-protocol, we know that for every $x \notin L$ and every crs_{Π}, an accepting $\widetilde{\Pi}$-transcript $(\mathsf{crs}_{\Pi}, k, x, \mathbf{a}, \mathbf{e}, \mathbf{z})$ must satisfy the condition that $h(k, \mathbf{a}) = \mathbf{e} = f(\mathsf{crs}_{\Pi}, x, \mathbf{a})$.

Suppose that some efficient prover P^*, given $x \notin L$ and a random $\mathsf{crs} = (\mathsf{crs}_{\Pi}, k)$, could find $(\mathbf{a}, \mathbf{e}, \mathbf{z})$ making the transcript $(\mathsf{crs}_{\Pi}, k, x, \mathbf{a}, \mathbf{e}, \mathbf{z})$ accepting with non-negligible probability. Then, by CRS indistinguishability, the same would be true for crs_{Π} sampled by the algorithm $\mathsf{TrapGen}(1^{\lambda}, x, \mathsf{aux})$ for an auxiliary input aux satisfying the correctness property of Definition 3.51. In other words, for $(\mathsf{crs}_{\Pi}, \tau_{\Pi}) \leftarrow \mathsf{TrapGen}(1^{\lambda}, x, \mathsf{aux})$ and $\mathsf{crs} = (\mathsf{crs}_{\Pi}, k)$, $P^*(x, \mathsf{crs})$ would output (with non-negligible probability) some $\mathbf{a}$ such that $h(k, \mathbf{a}) = f(\mathsf{crs}_{\Pi}, x, \mathbf{a}) = \mathsf{BadChallenge}(\tau_{\Pi}, \mathsf{crs}_{\Pi}, x, \mathbf{a})$.

This directly contradicts the correlation intractability of $\mathcal{H}$ for the relation $R_{\tau_{\Pi}, \mathsf{crs}_{\Pi}, x} = \{(\mathbf{a}, \mathbf{e}) : \mathbf{e} = \mathsf{BadChallenge}(\tau_{\Pi}, \mathsf{crs}_{\Pi}, x, \mathbf{a})\}$. In more detail, a correlation-intractability adversary $\mathcal{A}$ could break the correlation intractability of $\mathcal{H}$ by sampling $(\mathsf{crs}_{\Pi}, \tau_{\Pi})$ itself, declaring the relation $R_{\tau_{\Pi}, \mathsf{crs}_{\Pi}, x}$ to be broken, and then running $P^*(x, \mathsf{crs})$ after being given $k \leftarrow \mathcal{H}.\mathsf{Gen}(1^{\lambda})$. Since Π originally had $2^{-\lambda^{\epsilon}}$ soundness, the relation

$R_{\tau_\Pi,\mathsf{crs}_\Pi,x}$ indeed has subexponential sparsity, so this contradicts our assumption on $\mathcal{H}$. Thus, we conclude that $\mathcal{H}$ soundly instantiates the Fiat-Shamir heuristic for Π, as desired. □

3.6.2 Examples and Implications

It is easy to see that the variant of the [FLS90] Hamiltonicity protocol described in Section 3.5 satisfies Definition 3.50; the (instance-independent) trapdoor generation algorithm simply samples $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{PKE.Gen}(1^\lambda)$ and outputs pk as the common reference string and sk as the trapdoor. As already described, the bad-challenge function associated to Π_{FLS} is indeed efficiently computable given sk. Similarly, variants of the [Blu86] Hamiltonicity protocol in which the commitment scheme is instantiated using public-key encryption also satisfy Definition 3.50.[14]

We now describe an interesting example of an instance-dependent trapdoor Σ-protocol with a trapdoor that actually depends on the instance: the [GMR85] protocol for quadratic residuosity. Recall that an input $x = (N, y)$ to this protocol consists of an integer $N = pq$ that is a product of two primes along with an element $y \in \mathbb{Z}_N^\times$. An instance x is in the language $\mathbb{QR}$ if y is a quadratic residue modulo N. A witness w for this fact is a square root of y modulo N. The [GMR85] protocol $\Pi = \Pi_{\mathrm{GMR}}$ is described in Fig. 3-3.

$$\begin{array}{lcl} P(N, w) & & V(y) \\ r \leftarrow \mathbb{Z}_N^\times & \xrightarrow{\qquad a \qquad} & \\ a = r^2 & & \\ & \xleftarrow{\qquad e \qquad} & e \leftarrow \{0,1\} \\ z = rw^e & \xrightarrow{\qquad z \qquad} & \text{If } z^2 = Ay^e\text{, accept.} \end{array}$$

Figure 3-3: The Zero Knowledge Proof System Π_{GMR} for Quadratic Residuosity.

We additionally consider the protocol Π^t_{GMR} repeated t times in parallel for $t = \Omega(\lambda^\epsilon)$. Note that this is indeed a Σ-protocol (with an empty common reference

[14] This requires one further modification: the prover must additionally commit to the hidden permutation π and reveal it when asked to reveal the entire graph. We require this so that the bad-challenge function is computable given the PKE secret key – naively, the bad-challenge function would require solving a graph isomorphism problem.

string) with bad-challenge function $f(x, \mathbf{a}) = \mathbf{e}$ such that $e_i = QR(N, a_i)$ for all i, and $QR(N, a)$ is defined to be 1 if and only if a is a square mod N. This holds because for any $x = (N, y)$ such that y is not a quadratic residue modulo N, if $a \in \mathbb{Z}_N^\times$ and $QR(N, a) = 1$, then $QR(N, ay) = 0$ and hence then "1" challenge associated to a cannot be answered by any third message z; similarly, if $QR(N, a) = 0$ then the "0" challenge associated to a cannot be answered by any third message z.

Finally, we note that the function $f(x, \mathbf{a})$ is efficiently computable given the factorization of $N = p \cdot q$, so we conclude that Π_{GMR}^t is an instance-dependent trapdoor Σ-protocol with auxiliary information $\mathsf{aux} = (p, q)$ and trapdoor $\tau = \mathsf{aux}$ (satisfying subexponential soundness if $t \geq \lambda^\epsilon$). Thus, we conclude

Corollary 3.53. *Assuming the existence of circular-secure FHE, for any $t \geq \lambda^\epsilon$, there exists a hash family $\mathcal{H}$ soundly instantiating the Fiat-Shamir heuristic for Π_{GMR}^t.*

We obtain Corollary 3.5 as a consequence of Corollary 3.53 along with one of the main results from [DNRS99] (additionally assuming that $\mathsf{QR} \notin \mathsf{BPP}$), which generalizes to any protocol (not just Π_{GMR}^t):

Theorem 3.54 ([DNRS99])**.** *If there exists a hash family $\mathcal{H}$ that soundly instantiates the Fiat-Shamir transform for a 3-message protocol Π for a language $L \notin \mathsf{BPP}$, then Π is not zero knowledge.*

For clarity, we include a proof of Theorem 3.54; we only consider the standard definition of (auxiliary input) zero knowledge as opposed to the weakenings introduced in [DNRS99], but the argument extends to such weakenings.

Proof. (sketch) Let $\mathcal{H}$ soundly instantiate the Fiat-Shamir transform for Π, and suppose for the sake of contradiction that Π is zero knowledge. We the define the following cheating verifier V^* for Π:

- Auxiliary input: a hash key k (sampled according to $\mathcal{H}.\mathsf{Gen}$).

- Second message: upon receiving the first message $\mathbf{a}$, V^* computes $\mathbf{e} = h(k, \mathbf{a})$ and sends $\mathbf{e}$ to the prover.

- Output: upon receiving a third message $\mathbf{z}$, V^* outputs the transcript $(\mathbf{a}, \mathbf{e}, \mathbf{z})$.

If Π is (auxiliary input) zero knowledge, then there is a PPT simulator $S^*(x, k)$ that produces a computationally indistinguishable transcript $(\mathbf{a}, \mathbf{e}, \mathbf{z})$ (for all $x \in L$ and for $k \leftarrow \mathcal{H}.\mathsf{Gen}(1^\lambda)$). In particular, this guarantees that transcripts output by S^* satisfy (with all but negligible probability)

1. $\mathbf{e} = H_k(\mathbf{a})$, and
2. $V(x, \mathbf{a}, \mathbf{e}, \mathbf{z}) = 1$,

because these are both efficiently checkable conditions given (x, k).

Now, since we assumed that $L \notin \mathsf{BPP}$, there must *also* exist some $x^* \notin L$ such that $S^*(x^*, k)$ produces a transcript $(\mathbf{a}, \mathbf{e}, \mathbf{z})$ satisfying conditions (1) and (2) with all but negligible probability; otherwise, we would have a BPP algorithm deciding the language L (sample k, run $S^*(x, k)$, and check if conditions (1) and (2) are satisfied).

This allows us to contradict the soundness of the Fiat-Shamir protocol $\widetilde{\Pi}$ defined by Π and $\mathcal{H}$: the simulator S^*, given x^* and $k \leftarrow \mathcal{H}.\mathsf{Gen}(1^\lambda)$, exactly breaks the soundness of $\widetilde{\Pi}$ by the previous paragraph. This completes the proof. $\square$

Theorem 3.54 and Corollary 3.53 directly imply that the (parallel repeated) [GMR85] protocol is not zero-knowledge (assuming the existence of circular-secure FHE and the hardness of quadratic residuosity), as desired.

Part II

CI Self-Reductions and Further Applications to Protocols

Chapter 4

Fiat-Shamir for Repeated Squaring and Applications to PPAD-Hardness and VDFs

4.1 Introduction

The Fiat-Shamir transform [FS87] is a methodology for compiling a public-coin interactive proof (or argument) system for a language L into a non-interactive argument system for L. While originally developed in order to convert 3-message identification schemes into signature schemes, the methodology readily generalized [BR93] to apply to a broad, expressive class of interactive protocols, with applications including non-interactive zero knowledge for **NP** [BR93], succinct non-interactive arguments for **NP** [Mic94, BCS16], and widely used/practically efficient signature schemes [Sch89].

However, these constructions and results come with a big caveat: the security of the Fiat-Shamir transformation is typically *heuristic.* While the transformation has been proved secure (in high generality) in the random oracle model [BR93, PS96, Mic94, BCS16], it is known that some properties that hold in the random oracle model – including the soundness of Fiat-Shamir for certain contrived interactive arguments – cannot be instantiated at all in the standard model [CGH98, DNRS99, Bar01, GK03, BBH+19].

Given these negative results, security in the random oracle model is by no means the end of the story. Indeed, the question of whether Fiat-Shamir can be instantiated for any given interactive argument system (and under what computational assumptions this can be done) has been a major research direction over the last twenty years [DNRS99, Bar01, GK03, BLV03, CCR16, KRR17, CCRR18, HL18, CCH+19, PS19, BBH+19, BFJ+20, JJ19, LVW19]. After much recent work, some positive results are known, falling into three categories (in the decreasing order of strength of assumptions required):

1. We can compile *arbitrary* (constant-round, public-coin) interactive proofs under extremely strong assumptions [KRR17, CCRR18] that are *non-falsifiable* in the sense of [Nao03].

2. We can compile certain succinct interactive proofs [LFKN90, GKR08] – and variants of other interactive proofs not captured in item (3) below, such as [GMW86] – under extremely strong but *falsifiable* assumptions [CCH+19].

3. We can compile variants of some classical 3-message zero knowledge proof systems [GMR85, Blu86, FLS90] under *standard* cryptographic assumptions [CCH+19, PS19].

Elaborating on item (2) above, what is currently known is that the sumcheck protocol [LFKN90] and the related Goldwasser-Kalai-Rothblum (GKR) [GKR08] interactive proof system can be compiled under an "optimal security assumption" related to (secret-key) Regev encryption. Roughly speaking, an optimal hardness assumption is the assumption that some search problem cannot be solved with probability significantly better than repeatedly guessing a solution at random. This is an extremely strong assumption that (in the context of Regev encryption) requires careful parameter settings to avoid being trivially false.

In this work, we focus on improving item (2); in particular, we ask:

Under what computational assumptions can we instantiate Fiat-Shamir *for an interesting* succinct *interactive proof?*

Instead of considering the [LFKN90, GKR08] protocols, we work on compiling a protocol of Pietrzak [Pie18] for the "repeated-squaring language" [RSW96]. At a high level, Pietrzak constructs a "sumcheck-like" succinct interactive proof system for the computation $f_{N,g}(T) = g^{2^T} \pmod{N}$ over an RSA modulus $N = pq$. Compiling this protocol turns out to have applications related to verifiable delay functions (VDFs) [BBBF18] and hardness in the complexity class **PPAD** [CHK+19a,CHK+19b, EFKP19], which we elaborate on below.

Applications. We consider two apparently different questions: the first is that of establishing the hardness of the complexity class **PPAD** ("polynomial parity arguments on directed graphs") [Pap94] that captures the hardness of finding Nash equilibria in bimatrix games [DGP06, CDT09]; the second is that of constructing verifiable delay functions (VDFs), a recently introduced cryptographic primitive [BBBF18] which gives us a way to introduce delays in decentralized applications such as blockchains.

The Hardness of PPAD. Establishing the hardness of **PPAD** [Pap94], possibly under cryptographic assumptions, is a long-standing question in the foundations of cryptography and computational game theory. After two decades of little progress on the question, a recent sequence of works [BPR15, HY17, CHK+19a, CHK+19b, EFKP19] has managed to prove that there are problems in **PPAD** (and indeed a smaller complexity class, **CLS** [DP11]) that are hard (even *on average*) under *strong* cryptographic assumptions. The results so far fall roughly into two categories, depending on the techniques used.

1. **Program Obfuscation.** Bitansky, Paneth and Rosen [BPR15], inspired by an approach outlined in [AKV04], showed that **PPAD** is hard assuming the existence of subexponentially secure indistinguishability obfuscation (IO) [BGI+01, GGH+13] and one-way functions. This was later improved [GPS16, HY17] to rely on polynomially-secure functional encryption and to give hardness in **CLS** $\subseteq$ **PPAD**.

2. **Unambiguously Sound Incrementally Verifiable Computation.** The recent beautiful work [CHK$^+$19a] constructs a hard-on-average **CLS** instance assuming the existence of a special kind of incrementally verifiable computation (IVC) [Val08]. Instantiating this approach, they show that $\mathbf{CLS} \subseteq \mathbf{PPAD}$ is hard-on-average if there exists a hash function family that soundly instantiates the Fiat-Shamir heuristic [FS87] for the sumcheck interactive proof system for $\#\mathsf{P}$ [LFKN90]. Two follow-up works [CHK$^+$19b, EFKP19] show the same conclusion if Fiat-Shamir for Pietrzak's interactive proof system [Pie18] can be soundly instantiated (and if the underlying "repeated squaring language" is hard).

Regarding the first approach [BPR15, GPS16, HY17], secure indistinguishability obfuscators have recently been constructed based on the veracity of a number of non-standard assumptions (see, e.g., [AJL$^+$19, BDGM20a]). Regarding the second approach [CHK$^+$19a, CHK$^+$19b, EFKP19], the hash function can be instantiated in the random oracle model, or under "optimal KDM-security" assumptions [CCRR18, CCH$^+$19].

In summary, despite substantial effort, there are no known constructions of hard **PPAD** instances from standard cryptographic assumptions (although see Section 4.1.3 for a recent independent work [KPY20] that shows such a result under a new assumption on bilinear groups).

Verifiable Delay Functions. A Verifiable Delay Function (VDF) [BBBF18] is a function f with the following properties:

- f can be evaluated in some (moderately large) time T.
- Computing f (on average) requires time close to T, *even given a large amount of parallelism.*
- There is a time $T + o(T)$ procedure that computes $y = f(x)$ on an input x along with a proof π that $y = f(x)$ is computed correctly. This proof (argument)

system should be verifiable in time $\ll T$ (ideally $\mathsf{poly}(\lambda, \log T)$)) and satisfy standard (computational) soundness.

Since their introduction [BBBF18], there have been a few proposed candidate VDF constructions [BBBF18, Pie18, Wes19, dFMPS19, EFKP19]. There are currently no constructions based on standard cryptographic assumptions, but this is somewhat inherent to the primitive: a secure VDF implies the existence of a problem which can be solved in time T and also requires (sequential) time close to T. Nonetheless, one can ask[1] whether VDFs can be constructed from "more standard-looking" assumptions, a question partially answered by [Pie18, Wes19]. In particular, each of their constructions relies on two assumptions:

(1) The T-repeated squaring problem [RSW96] requires sequential time close to T.

(2) The Fiat-Shamir heuristic for some specific public-coin interactive proof/argument[2] can be soundly instantiated.

The techniques used in both the construction of hard **PPAD** instances and the construction of VDFs are similar, and so are the underlying assumptions (this is due to the connection between **PPAD** and incrementally verifiable computation [Val08, CHK$^+$19a]). In particular, the works of [CHK$^+$19b, EFKP19] construct hard **PPAD** (and even **CLS**) instances under two assumptions:

(1′) The T-repeated squaring problem [RSW96] requires super-polynomial (standard) time for some $T = \lambda^{\omega(1)}$.

(2′) The Fiat-Shamir heuristic for a variant of the [Pie18] interactive proof system can be soundly instantiated.

The assumption (1) (and its weakening, assumption (1′)) is the foundation of the Rivest-Shamir-Wagner time-lock puzzle [RSW96] and has been around for over 20

[1] [BBBF18] explicitly suggested this.

[2] The two works [Pie18, Wes19] consider qualitatively different interactive argument systems. In this work, we focus on the [Pie18] protocol since (1) it has unconditional soundness and therefore is more conducive to provable Fiat-Shamir compilation, and (2) it is more closely related to **PPAD**-hardness.

years. In particular, breaking the RSW assumption has received renewed cryptanalytic interest recently [Riv99, Fab19].

On the other hand, as previously discussed, the assumptions $(2, 2')$ are not well understood. Indeed, our main question about Fiat-Shamir for succinct arguments (if specialized to the [Pie18] protocol) is intimately related to the following question.

Can we construct hard **PPAD** *instances and VDFs under more well-studied assumptions?*

4.1.1 Our Results

We show how to instantiate the Fiat-Shamir heuristic for the [Pie18] protocol under a quantitatively strong (but relatively standard) variant of the Learning with Errors (LWE) assumption [Reg05]. We give a family of constructions of hash functions that run in subexponential (or even quasi-polynomial or polynomial) time, and prove that they soundly instantiate Fiat-Shamir for this protocol under a sufficiently strong LWE assumption.

More generally, we extend the "bad-challenge function" methodology of [CCH+19] for proving the soundness of Fiat-Shamir to a class of protocols whose bad-challenge functions are not efficiently computable. We elaborate on this below in the technical overview (Section 4.1.4).

As a consequence, we obtain **CLS**-hardness and VDFs from a pair of quantitatively related assumptions on the [RSW96] repeated squaring problem and on the learning with errors (LWE) problem [Reg05]; the latter can in turn be based on the worst-case hardness of the (approximate) shortest vector problem (GapSVP) on lattices. In particular, we can base the hardness of $\mathbf{CLS} \subseteq \mathbf{PPAD}$, as well as the security of a VDF, on the hardness of two relatively well-studied problems.

Fiat-Shamir for Pietrzak's Protocol. For our main result, we show that for any $\epsilon > 0$, an LWE assumption of quantitative strength $2^{n^{1-\epsilon}}$ allows for a Fiat-Shamir instantiation with verification runtime $2^{\tilde{O}(n^{\epsilon})}$ on a repeated squaring instance with security parameter $\lambda = O(n \log n)$. Such a result is meaningful as long as the verification runtime is smaller than the time it takes to solve the repeated squaring problem; the current best known algorithms for repeated squaring run in heuristic time $2^{\tilde{O}(\lambda^{1/3})} = 2^{\tilde{O}(n^{1/3})}$ [LLMP90].

Here and throughout the paper, we will use (t, δ)-hardness to denote that a cryptographic problem is hard for t-time algorithms to solve with δ probability (or distinguishing advantage).

Theorem 4.1. *Let $\epsilon > 0$ be arbitrary. Assume that (decision)* LWE *is* $\left(2^{\tilde{O}(n^{1/2})}, 2^{-n^{1-\epsilon}}\right)$*-hard (or alternatively,* $\left(2^{\tilde{O}(n^{\epsilon})}, 2^{-n^{1-\epsilon}}\right)$*-hard for non-uniform algorithms). Then, there exists a hash family $\mathcal{H}$ that soundly instantiates the Fiat-Shamir heuristic for Pietrzak's interactive proof system [Pie18]. When the proof system is instantiated for repeated squaring over groups of size $2^{O(\lambda)}$ with $\lambda = O(n \log n)$, the hash function h from the family $\mathcal{H}$ can be evaluated in time $2^{\tilde{O}(\lambda^{\epsilon})}$.*

Under the assumption that (decision) LWE *is* $\left(2^{\tilde{O}(n^{1/2})}, 2^{-\frac{n}{\log^c n}}\right)$*-hard for some constant $c > 0$ (or alternatively,* $\left(\mathsf{quasipoly}(n), 2^{-\frac{n}{\log^c n}}\right)$*-hard for non-uniform algorithms), there exists such a hash family $\mathcal{H}$ with quasi-polynomial evaluation time.*

Moreover, the LWE assumption that we make falls into the parameter regime where we know worst-case to average-case reductions [Reg05, BLP+13, PRSD17], so we obtain the following corollary.

Corollary 4.2. *The conclusions of Theorem 4.1 (with parameter $\epsilon < \frac{1}{2}$) follow from the assumption that the worst case problem* $\mathsf{poly}(n)$-GapSVP *for rank n lattices requires time $2^{\omega(n^{1-\epsilon})}$. Similarly, the protocol with quasi-polynomial verification time is sound under the assumption that* $\mathsf{poly}(n)$-GapSVP *requires time $2^{\frac{n}{\log(n)^c}}$ for some $c > 0$.*

The Shortest Vector Problem (SVP) on integer lattices is a well-studied problem (see discussion in [Pei16, ADRS15]); despite a substantial effort, all known $\mathsf{poly}(n)$-approximation algorithms for the problem have exponential run-time $2^{\Omega(n)}$. As a result, our current understanding of the approximate-SVP landscape is consistent with the following conjecture.

Conjecture 1 (Exponential Time Hypothesis for GapSVP)**.** *For any fixed* $\gamma(n) = \mathsf{poly}(n)$*, the* $\gamma(n)$-GapSVP *problem cannot be solved in time* $2^{o(n)}$*.*

Assuming Conjecture 1, the conclusion of Theorem 4.1 holds for every $\epsilon > 0$; moreover, the variant of the Theorem 4.1 protocol with quasi-polynomial time evaluation is sound as well.

What about polynomial-time verification? Given a non-interactive protocol for repeated squaring with $2^{\tilde{O}(\lambda^\epsilon)}$ verification time (or quasi-polynomial evaluation time), one can always define a new security parameter $\kappa = 2^{\tilde{O}(\lambda^\epsilon)}$ (or $\kappa = 2^{\log(\lambda)^c}$) to obtain a protocol with *polynomial-time* verification. However, this makes use of *complexity leveraging* [CGGM00], so (i) this requires making the assumption that repeated squaring (on instances with security parameter λ) is hard for $\mathsf{poly}(\kappa(\lambda))$-time adversaries, and (ii) the resulting protocol cannot have security subexponential in κ.

If one does not wish to use complexity leveraging, we give an alternative construction that has (natively) polynomial-time verification, at the cost of a stronger LWE assumption.

Theorem 4.3. *Let* $\delta > 0$ *be arbitrary and* $q(n) = \mathsf{poly}(n)$ *be a fixed (sufficiently large) polynomial in* n*. Assume that (decision)* LWE *is* $\left(\mathsf{poly}(n), q^{-\delta n}\right)$*-hard for non-uniform distinguishers (or* $\left(2^{\tilde{O}(n^{1/2})}, q^{-\delta n}\right)$*-hard for uniform distinguishers). Then, there exists a hash family* $\mathcal{H}$ *that soundly instantiates the Fiat-Shamir heuristic for Pietrzak's interactive proof system [Pie18] with* $\mathsf{poly}(\lambda) = \mathsf{poly}(n \log n)$*-time verification. More specifically, the verification time is* $\lambda^{O(1/\delta)}$*.*

Moreover, this strong LWE assumption *still* falls into the parameter regime with a meaningful worst-case to average-case reduction:

Corollary 4.4. *The conclusion of Theorem 4.3 follows from the assumption that worst-case* $\gamma(n)$-GapSVP *(for a fixed* $\gamma(n) = \mathsf{poly}(n)$*) cannot be solved in time* $n^{o(n)}$ *with* $\mathsf{poly}(n)$ *space and* $\mathsf{poly}(n)$ *bits of nonuniform advice (independent of the lattice).*

Polynomial-space algorithms for GapSVP have themselves been an object of study for over 25 years [Kan83, KF16, BLS16, ABF+20], but the current best (poly-space) algorithms for this problem run in time $n^{\Omega(\epsilon n)}$ for approximation factor $n^{1/\epsilon}$. Therefore, under a sufficiently strong (and plausible) worst-case assumption about GapSVP, we have a polynomial-time Fiat-Shamir compiler without complexity leveraging.

By combining Theorems 4.1 and 4.3 with the results of [CHK+19b, EFKP19], we obtain the following construction of hard-on-average **CLS** instances.

Theorem 4.5. *For a constant $\epsilon > 0$, suppose that*

- *n-dimensional* LWE *(with polynomial modulus) is $\left(2^{\tilde{O}(n^{1/2})}, 2^{-n^{1-\epsilon}}\right)$-hard, and*
- *The repeated squaring problem on an instance of size 2^λ requires $2^{\lambda^\epsilon \log(\lambda)^{\omega(1)}}$ time.*

Then, there is a hard-on-average problem in **CLS** $\subseteq$ **PPAD**. *The same conclusion holds if for some $c > 0$,*

- LWE *is $\left(2^{\tilde{O}(n^{1/2})}, 2^{-\frac{n}{\log(n)^c}}\right)$-hard, and*
- *The repeated squaring problem is hard for quasi-polynomial time algorithms.*

The same conclusion also holds if for some $\delta > 0$,

- LWE *is $\left(\mathsf{poly}(n), q^{-\delta n}\right)$-hard for non-uniform distinguishers, and*
- *The repeated squaring problem is hard for polynomial time algorithms.*

We obtain Theorem 4.5 by plugging our standard model Fiat-Shamir instantiation into the complexity-theoretic reduction of [CHK+19b].[3] For use in this reduction, our non-interactive protocol must satisfy a stronger security notion called *(adaptive) unambiguous soundness* [RRR16, CHK+19a], which we show is indeed the case.

Note that the two hardness assumptions in the theorem statement are in opposition to each other. As ϵ becomes smaller, the repeated squaring assumption becomes weaker, but the LWE assumption becomes stronger. In particular, we cannot

[3] Our protocol differs very slightly from the formulation in [CHK+19b], but the difference is irrelevant to the reduction.

set $\epsilon \geq 1/3$ as there are known algorithms [LLMP90] solving repeated squaring in (heuristic) time $2^{\widetilde{O}(\lambda^{1/3})}$.

Additionally, as a direct consequence of Theorem 4.1, we obtain VDFs in the standard model as long as the underlying repeated squaring problem is sufficiently (sequentially) hard. Recall that the repeated squaring problem [RSW96] is the computation of the function $f_{N,g}(T) = g^{2^T} \pmod N$, for the appropriate distribution on $N = pq$ and g.

Theorem 4.6. *For a constant $\epsilon > 0$, suppose that*

- LWE *is $\left(2^{\tilde{O}(n^{1/2})}, 2^{-n^{1-\epsilon}}\right)$-hard, and*
- *The repeated squaring problem [RSW96] over groups of size $2^{O(\lambda)}$ requires $T(1-o(1))$ sequential time for $T \gg 2^{\tilde{O}(\lambda^\epsilon)}$.*

Then, the repeated squaring function $f_{N,g}$ can be made into a VDF with verification time $2^{\tilde{O}(\lambda^\epsilon)}$ on groups of size $2^{O(\lambda)}$ (with $\lambda = O(n \log n)$). Similarly, if for some $c > 0$,

- LWE *is $\left(2^{\tilde{O}(n^{1/2})}, 2^{-\frac{n}{\log(n)^c}}\right)$-hard, and*
- *The repeated squaring problem requires $T(1-o(1))$ sequential time for $T \gg 2^{\tilde{O}(\log(\lambda)^{c+1})}$,*

Then, $f_{N,g}$ can be made into a VDF with verification time $2^{\tilde{O}(\log(\lambda)^{c+1})}$. Finally, if for some $\delta > 0$,

- LWE *(with modulus q) is $\left(\mathsf{poly}(n), q^{-\delta n}\right)$-hard for non-uniform distinguishers, and*
- *The repeated squaring problem requires $T(1-o(1))$ sequential time for all $T = \mathsf{poly}(\lambda)$.*

Then, $f_{N,g}$ can be made into a VDF with $\lambda^{O(1/\delta)}$-time verification.

Theorem 4.6 follows immediately from Theorem 4.1 along with the construction of Pietrzak [Pie18]. While many of the VDFs in Theorem 4.6 have super-polynomial verification time (and therefore do not fit the standard definition), they can be converted into (standard) VDFs with polynomial verification time via complexity leveraging; however, the leveraged VDFs will only support quasi-polynomial (respectively, $2^{2^{\mathsf{poly}\log\log\kappa}}$) time computation (and soundness of the VDF will only hold against adversaries running in time quasi-polynomial in the new security parameter κ). Because of this, we consider the formulation in terms of super-polynomial time verification to be more informative.

4.1.2 Comparison with Prior Work

Cryptographic Hardness of PPAD. As described in the introduction, prior works on the cryptographic hardness of **PPAD** fall into two categories – those based on obfuscation and ones based on incrementally verifiable computation (IVC). The obfuscation-based constructions all make cryptographic assumptions related to the existence of indistinguishability obfuscation or closely related primitives that we currently do not know how to instantiate based on well-studied assumptions. (For the latest in obfuscation technology, we refer the reader to [JLMS19, JLS19].) We therefore focus on comparing to the previous IVC-based constructions.

- [CHK$^+$19a] constructs hard problems in **CLS** under the polynomial hardness of #SAT with poly-logarithmically many variables along with the assumption that Fiat-Shamir can be soundly instantiated for the sumcheck protocol [LFKN90]. The latter follows either in the random oracle model or under the assumption that a LWE-based fully homomorphic encryption scheme is "optimally circular-secure" [CCH$^+$18, CCH$^+$19] for quasi-polynomial time adversaries.

  While the hardness of #SAT (with this parameter regime) is a weaker assumption than the subexponential hardness of repeated squaring, the [CHK$^+$19a] (standard model) result has the drawback of relying on an optimal hardness assumption. Roughly speaking, an optimal hardness assumption is the assumption that some search problem cannot be solved with probability significantly better than repeatedly guessing a solution at random. This is an extremely strong assumption that requires careful parameter settings to avoid being trivially false.

In contrast, our main LWE assumption is *subexponential* (concerning distinguishing advantage $2^{-n^{1-\epsilon}}$) and follows from the worst-case hardness of $\mathsf{poly}(n)$-GapSVP for time $2^{n^{1-\epsilon}}$ algorithms. Even our most optimistic LWE assumption (as in Theorem 4.3) follows from a form of worst-case hardness quantitatively far from the corresponding best known algorithms.

- [CHK$^+$19b,EFKP19] construct hard problems in **CLS** assuming the polynomial hardness of repeated squaring along with a generic assumption that the Fiat-Shamir heuristic can be instantiated for round-by-round sound (see [CCH$^+$18, CCH$^+$19]) public-coin interactive proofs. The latter can be instantiated either in the random oracle model, or under the assumption that Regev encryption (or ElGamal encryption) is "optimally KDM-secure" for unbounded KDM functions [CCRR18].

  The [CCRR18] assumption is (up to minor technical details) stronger than the optimal security assumption used in [CHK$^+$19a] (because the security game additionally involves an unbounded function), so the [CHK$^+$19b, EFKP19] are mostly framed in the random oracle model. In this work, we give a new Fiat-Shamir instantiation to plug into the [CHK$^+$19b, EFKP19] framework.

VDFs. We compare our construction of VDFs to previous constructions [BBBF18, Pie18, Wes19, dFMPS19, EFKP19].

- [BBBF18] and [dFMPS19] give constructions of VDFs from new cryptographic assumptions related to permutation polynomials and isogenies over supersingular elliptic curves, respectively. These assumptions are certainly incomparable to ours, but we rely on the hardness of older, more well-studied problems.

- [Pie18, EFKP19] have the same basic VDF construction as ours; the main difference is that they use a random oracle to instantiate their hash function, while we use a hash function in the standard model and prove its security under a quantitatively strong variant of LWE.

- [Wes19] also builds a VDF based on the hardness of repeated squaring, but by building a different interactive argument for computing the function and assuming that Fiat-Shamir can be instantiated for this argument. Again, this assumption holds in the random oracle model, but we know of no instantiation of this VDF in the standard model.

On the negative side, our main VDF (for the natural choice of security parameter) has verification time $2^{\tilde{O}(\lambda^\epsilon)}$; this can be thought of as polynomial-time via complexity leveraging, but this results in a VDF that is only quasi-polynomially secure. Alternatively, based on our optimistic LWE assumption, we only obtain a VDF with large polynomial (i.e. $\lambda^{1/\delta}$ for small δ) verification time. As a result, we consider our VDF construction to be a proof-of-concept regarding whether VDFs can be built based on "more standard-looking assumptions", in particular, without invoking the random oracle model.

4.1.3 Additional Related Work

[BG20] constructs hard instances in the complexity class **PLS** – which contains **CLS** and is incomparable to **PPAD** – under a falsifiable assumption on bilinear maps introduced in [KPY19] (along with the randomized exponential time hypothesis (ETH)).

In recent independent work, [KPY20] constructs hard-on-average **CLS** instances under the (quasi-polynomial) [KPY19] assumption. In fact, they give a protocol for unambiguous and incrementally verifiable computation for all languages decidable in space-bounded and slightly super-polynomial time.

4.1.4 Technical Overview

We now discuss the ideas behind our main result, Theorem 4.1, which is an instantiation of the Fiat-Shamir heuristic for the [Pie18] repeated squaring protocol. In obtaining this result, we also broaden the class of interactive proofs for which we have Fiat-Shamir instantiations under standard assumptions.

The main tool used by our construction is a hash function family $\mathcal{H}$ that is correlation intractable [CGH98] for *efficiently computable functions* [CLW18, CCH+19]. Recall that a hash family $\mathcal{H}$ is correlation intractable for t-time computable functions if for every function f computable time t, the following computational problem is hard: given a description of a hash function h, find an input x such that $h(x) = f(x)$. We now know [PS19] that such hash families can be constructed under the LWE assumption.

Correlation Intractability and Fiat-Shamir. In order to describe our result, we first sketch the [CCH+19] paradigm for using such a hash family $\mathcal{H}$ to instantiate the Fiat-Shamir heuristic.

For simplicity, consider a three-message (public-coin) interactive proof system (Σ-protocol)

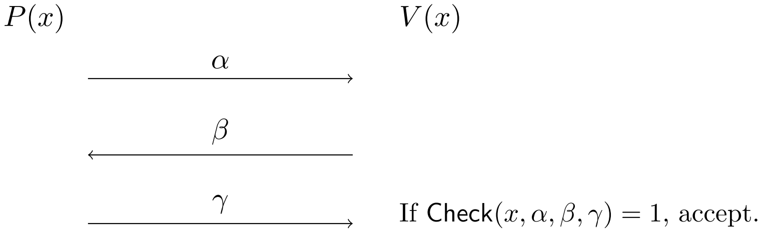


Figure 4-1: A Σ-protocol Π.

as well as its corresponding Fiat-Shamir round-reduced protocol $\Pi_{\mathrm{FS},\mathcal{H}}$ for a hash family $\mathcal{H}$.

Moreover, suppose that this protocol Π satisfies the following soundness property (sometimes referred to as "special soundness"): for every $x \notin L$ and every prover message α, there exists at most one verifier message $\beta^*(x, \alpha)$ allowing the prover to

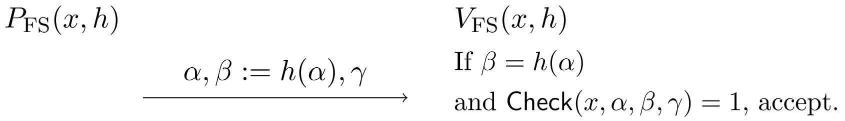

$P_{\mathrm{FS}}(x,h)$
$V_{\mathrm{FS}}(x,h)$
$\alpha, \beta := h(\alpha), \gamma$
If $\beta = h(\alpha)$
and $\mathsf{Check}(x,\alpha,\beta,\gamma) = 1$, accept.

Figure 4-2: The Protocol $\Pi_{\mathrm{FS},\mathcal{H}}$.

cheat.[4]

It then follows that if a hash family $\mathcal{H}$ is correlation intractable for the function family $f_x(\alpha) = \beta^*(x,\alpha)$, then $\mathcal{H}$ instantiates the Fiat-Shamir heuristic for Π.[5] This is because a cheating prover P^*_{FS} breaking the soundness of $\Pi_{\mathrm{FS},\mathcal{H}}$ must find a first message α such that its corresponding challenge $h(x,\alpha)$ is equal to the bad challenge $f_x(\alpha)$ (or else it has no hope of successfully cheating).

Therefore, using the hash family of [PS19], we can (under the LWE assumption) do Fiat-Shamir for any protocol Π whose "bad-challenge function" $f_x(\alpha)$ is computable in polynomial time; this has the important caveat that the complexity of computing the hash function h is at least the complexity of computing $f_x(\alpha)$.

This paradigm seems to run into the following roadblock: intuitively, for protocols Π of interest, computing $f_x(\alpha)$ appears to be *hard* rather than easy. For example,

1. For a standard construction of zero-knowledge proofs for **NP** such as [Blu86], computing $f_x(\alpha)$ involves breaking a cryptographically secure commitment scheme.

2. For (unconditional) statistical zero knowledge protocols such as the [GMR85] Quadratic Residuosity protocol, computing $f_x(\alpha)$ involves deciding the underlying hard language L.

3. For doubly efficient interactive proofs such as the [GKR08] interactive proof for logspace-uniform NC, computing $f_x(\alpha)$ again involves deciding the underlying language L; in this case, L is in P, but this Fiat-Shamir compiler would result in a non-interactive argument whose verifier runs in time longer than it takes to decide L.

[4] The prover can cheat on a pair (α,β) if and only if there *exists* a third message γ such that (x,α,β,γ) is accepted by the verifier.

[5] To obtain *adaptive soundness*, we modify the protocol to set $\beta = h(x,\alpha)$ and instead consider the function $f(x,\alpha) = \beta^*(x,\alpha)$.

The work [CCH$^+$19] resolves issues (1) and (2) in the following way: in both cases, we can arrange for $f_x(\alpha)$ to be efficiently computable *given an appropriate trapdoor*: in the case of [Blu86], the commitment scheme can have a trapdoor allowing for efficient extraction, while in the case of [GMR85], $f_x(\alpha)$ is efficient given an appropriate **NP**-witness for the complement language $\overline{L}$. However, we have no analogous resolution to (3), which is the setting of interest to us.[6]

The bad-challenge function of the [Pie18] protocol. With this context in mind, we now consider the [Pie18] protocol.[7] This protocol (like the [GKR08] protocol and the related sumcheck protocol [LFKN90]) is not a constant-round protocol, but is instead composed of up to polynomially many "reduction steps" of the following form.

$P(N = pq, T, g, h = g^T)$ $\qquad\qquad$ $V(N, T, g, h)$

Compute $u = g^{2^{T/2}}$ $\qquad \xrightarrow{\quad u \quad}$

$\qquad\qquad \xleftarrow{\quad r \quad}$

Compute $g' = u \cdot g^r, h' = h \cdot u^r$ $\qquad\qquad$ Compute $g' = u \cdot g^r, h' = h \cdot u^r$

Recurse on the statement $(N, T/2, g', h')$.

Figure 4-3: One reduction step of the [Pie18] protocol.

That is, the prover sends u, the (supposed) "halfway point" of the computation, yielding two derivative claims: $u = g^{2^{T/2}}$ and $h = u^{2^{T/2}}$. The verifier then challenges the prover to prove a random linear combination of the two statements: $h \cdot u^r = (u \cdot g^r)^{2^{T/2}}$.

Soundness can then be analyzed in a "round-by-round" fashion [CCH$^+$19]: if you start with a false statement (or if you start with a true statement but send an incorrect

[6] The only current known Fiat-Shamir instantiation for the [GKR08] protocol utilizes a *compact* correlation intractable hash family (in the sense that the hash evaluation time is independent of the time to compute the correlation function/relation) which we only know how to build from an optimal security assumption [CCH$^+$19].

[7] For this overview, we ignore the details of working over the group $\mathbb{QR}_N \subseteq \mathbb{Z}_N^\times$ and the corresponding technical challenges.

value $\tilde{u} \neq u$), there is at most one[8] bad challenge r^* resulting in a recursive call on a true statement.

To invoke the [CCH+19] paradigm, we ask: how efficiently can we compute the function $f(N, T, g, h, u) = r^*$? To answer this question, let $\tilde{g}$ denote a fixed group element of order $\phi(N)/2$ such that $g, h, u \in \langle \tilde{g} \rangle$. Letting γ, η, ω denote the discrete logs of g, h, and u in base $\tilde{g}$, we see that (for corresponding challenge r) the statement $(N, T/2, g', h')$ is true if and only if

$$\eta + r \cdot \omega \equiv 2^{T/2}(\omega + r \cdot \gamma) \pmod{\phi(N)/2}.$$

As a result, we see that r can be efficiently computed from the following information:

- The discrete logarithms η, ω, γ, and
- The factorization of N.

While the factorization of N can be known a priori in the security reduction (similar to prior work), the discrete logarithms depend on the prover message u and (adaptively chosen) statement (g, h). We conclude that the "bottleneck" for computing f is the problem computing a constant number of discrete logarithms in $\mathbb{Z}_p^\times$.

Since computing discrete logarithms over $\mathbb{Z}_p^\times$ is believed to be hard, and is not known to have a trapdoor, it appears unlikely that this approach would allow us to rely on the polynomial hardness of the [PS19] hash family. However, it *is* plausible that we could use a variant of the [PS19] hash family supporting *super-polynomial* time computation (proven secure under a super-polynomial variant of LWE) to capture the complexity of computing discrete logarithms.

Unfortunately, the naive version of this approach fails: the best known runtime bounds[9] for computing discrete logarithms over $\mathbb{Z}_p^\times$ for $p = 2^{O(\lambda)}$ are of the form $2^{\tilde{O}(\lambda^{1/2})}$ [Adl79, Pom87], and the best known heuristic algorithms (plausibly) run in time $2^{\tilde{O}(\lambda^{1/3})}$ [LLMP90]. If we were to instantiate the [PS19] hash family to support

[8] To guarantee this property, r is selected from a range smaller than either of the prime factors of N.

[9] See [JOP14] for a detailed discussion of the state-of-the-art on discrete logarithm algorithms.

functions of this complexity, we could prove the soundness of Fiat-Shamir for the [Pie18] protocol, but the resulting non-interactive protocol would run in time $2^{\tilde{O}(\lambda^{1/2})}$ (or in time $2^{\tilde{O}(\lambda^{1/3})}$ with a heuristic security proof); these are the same runtime bounds for the best known algorithms for solving the repeated squaring problem [Dix81, Pom87, LLMP90] (via factoring the modulus N). In other words, the verifier would run in enough time to be able to solve the repeated squaring problem itself. This is a very similar problem to issue (3) regarding the [LFKN90, GKR08] protocols, so we appear to be stuck.

Computing bad-challenge functions with low probability. We overcome the above problem with the following idea:

> What if we give up on computing the bad-challenge function *exactly*, and instead compute it using a *faster* randomized algorithm with *low success probability*?

In other words, we consider a new variant of the [CCH+19] framework for instantiating Fiat-Shamir in the standard model, where:

- An interactive protocol Π is characterized by some bad-challenge function f,
- f can be computed by a time t algorithm (or size s circuit) with some small but non-trivial probability δ.
- The hash function $\mathcal{H}$ is assumed to be correlation intractable – with sufficiently strong quantitative security – against adversaries running in time t (or with size s).

Then, it turns out that the resulting non-interactive protocol is sound! Informally, this is because if f is “approximated” by a time t-computable randomized function g_r (in the sense that $g_r(x)$ and $f(x)$ agree with probability δ on a worst-case input), then an adversary breaking the protocol $\Pi_{\mathrm{FS},\mathcal{H}}$ will break the correlation intractability of $\mathcal{H}$ with respect to g (rather than f) with probability δ. More formally, a cheating prover P^*_{FS} yields an algorithm that breaks the correlation intractability of $\mathcal{H}$ with respect to f, which in turn breaks the correlation intractability of $\mathcal{H}$ with respect to g_r (for hard-coded randomness r) with probability $\delta \cdot \frac{1}{\mathsf{poly}(\lambda)}$ (since g_r and f agree on an arbitrary input with probability at least δ). Therefore, if $\mathcal{H}$ is $(t, \delta \cdot \lambda^{-\omega(1)})$-secure, we conclude that $\Pi_{\mathrm{FS},\mathcal{H}}$ is sound.

This modification allows us to instantiate Fiat-Shamir for the [Pie18] protocol. In particular, we make use of folklore[10] [CCRR18] *preprocessing algorithms* for the discrete logarithm problem over $\mathbb{Z}_p^\times$ that run in time 2^{λ^ϵ} and have success probability $2^{-\lambda^{1-\epsilon}}$. More specifically, we consider a computation of the bad challenge function $f(N, T, g, h, u)$ in the following model:

- Hard-code (1) the factorization $N = pq$, (2) an appropriately chosen group element $\tilde{g}$ of high order, and (3) $2^{\tilde{O}(\lambda^\epsilon)}$ discrete logarithms (of fixed numbers modulo p and modulo q, respectively) in base $\tilde{g}$.
- Compute a (constant-size) collection of worst-case discrete logarithms by the standard index calculus algorithm [Adl79] in time $2^{\tilde{O}(\lambda^\epsilon)}$ with success probability $2^{-\lambda^{1-\epsilon}}$.

This can be thought of as either a non-uniform $2^{\tilde{O}(\lambda^\epsilon)}$-time algorithm, or a $2^{\tilde{O}(\lambda^\epsilon)}$-time algorithm with $2^{\tilde{O}(\lambda^{1/2})}$-time preprocessing.[11] By using this algorithm for the computation of the bad-challenge function $f(N, T, g, h, u)$, we obtain a Fiat-Shamir instantiation with verification time $2^{\tilde{O}(\lambda^\epsilon)}$ – a meaningful result as long as this runtime does not allow for solving the repeated squaring problem. Finally, the required assumption is that the [PS19] hash function is correlation intractable for adversaries that succeed with probability $2^{-\lambda^{1-\epsilon}}$, which holds under the claimed LWE assumption with parameters (n, q) for $\lambda = n \log q$.

Generalizations. In this overview, we focused specifically on the [Pie18] protocol, but our techniques give general blueprints for obtaining Fiat-Shamir instantiations.

[10] We are not aware of prior work considering this particular time-probability trade-off, but the necessary smooth number bounds appear in [CEP83, Gra08]. Quite curiously, [CCRR18] considers the $\mathsf{poly}(\lambda)$-time variant of this algorithm to give evidence against the optimal hardness of computing discrete logarithms over $\mathbb{Z}_p^\times$. That was bad for them, but for us, the non-optimal hardness is a feature!

[11] This second variant allows for an invocation of correlation intractability against uniform adversaries in the security proof.

We believe these blueprints may be useful in future work, so we state them (as "meta-theorems") explicitly here:

- **Fiat-Shamir for protocols with low success probability bad-challenge functions**. Our approach shows that if an interactive protocol Π is governed by a bad-challenge function f that is computable by an efficient randomized algorithm that is only correct with (potentially very) low probability, it is still possible to instantiate Fiat-Shamir for Π under a sufficiently strong LWE assumption.

- **Fiat-Shamir for discrete-log based bad-challenge functions**. Our approach also shows that if a protocol Π is governed by a bad-challenge function f that is efficiently computable given *oracle access*[12] to a *discrete log solver* (over $\mathbb{Z}_p^\times$ for $p \leq 2^{O(\lambda)}$), then it is possible to instantiate Fiat-Shamir for Π under a sufficiently strong LWE assumption.

We formalize both of these "meta-theorems" in the language of correlation intractability (rather than Fiat-Shamir) in Section 4.3.

Organization. The rest of the paper is organized as follows. Section 4.2 consists of the relevant preliminaries to describe and prove our results. In Section 4.3, we state and prove our results about low-success probability bad-challenge functions (and discrete-log based bad-challenge functions in particular) through the lens of correlation intractability. In Section 4.4, we formalize the round-by-round soundness property necessary to conclude the "adaptive unambiguous soundness" [CHK+19a] of the round-reduced [Pie18] protocol that suffices for **CLS**-hardness. In Section 4.5, we describe and analyze (our variant of) the [Pie18] protocol within the outlined framework and prove Theorem 4.1. Finally, in Section 4.6, we apply Theorem 4.1 to obtain Theorem 4.5 and Theorem 4.6.

[12] Crucially, we must also bound the number of calls that can be made to the oracle to be at most $\mathsf{poly}\log(\lambda)$ to get a meaningful result.

4.2 Preliminaries

4.2.1 Repeated Squaring modulo a Composite

Following [Pie18, CHK+19b], we consider the following formulation of the RSW time-lock puzzle [RSW96]. For an integer $N = pq$, recall that $\mathbb{Z}_N^\times$ is defined to be the group of units mod N, $\mathbb{QR}_N$ is defined to be the group of quadratic residues mod N, and $\mathbb{QR}_N^+$ is defined to be the set $\left\{x : 0 \leq x \leq \frac{N}{2} \text{ and } \left(\frac{x}{N}\right) = 1\right\}$, where $\left(\frac{\cdot}{N}\right)$ is defined to be the Jacobi symbol.

We now define (our variant of the) RSW moderately hard function.

- $\mathsf{Setup}(1^\lambda)$: On input the security parameter, sample an integer $N = pq$ along with a group element $\tilde{g} \in \mathbb{Z}_N^\times$ such that p, q are uniformly random safe primes in the range $[2^\lambda, 2^{\lambda+1}]$ and $\tilde{g}$ has order $\phi(N)/2$ in $\mathbb{Z}_N^\times$ (for example, $\tilde{g}$ can be the CRT lift of any generator for $\mathbb{Z}_p^\times$ and any generator for $\mathbb{Z}_q^\times$). Let $p' = \frac{p-1}{2}, q' = \frac{q-1}{2}$ (primes by construction), and note that $g := \tilde{g}^2$ generates $\mathbb{QR}_N$. Output (N, g).

- **Function evaluation**. Define the function

$$f_{N,g}(T) = g^{2^T} \pmod{N}.$$

We note that for any (N, g), the function $f_{N,g}(T)$ can be computed in time T.[13] We now consider two hardness assumptions related to the RSW moderately hard function.

Definition 4.7 ($t(\lambda)$-RSW Hardness Assumption)**.** *For some efficiently computable function $T(\cdot)$, computing $f_{N,g}(T(\lambda))$ for $(N, g) \leftarrow \mathsf{Setup}(1^\lambda)$ requires time $t(\lambda)$.*

For our main result on PPAD-hardness, we will assume the 2^{λ^ϵ}-RSW hardness assumption for some constant $\epsilon > 0$.

[13] As in prior work, we measure time complexity in terms of group operations.

Definition 4.8 ((σ, p)-RSW Sequentiality Assumption)**.** *For some efficiently computable function $T(\cdot)$, computing $f_{N,g}(T(\lambda))$ for $(N, g) \leftarrow \mathsf{Setup}(1^\lambda)$ requires $\sigma(T)$ sequential time for algorithms with $p(\lambda, T)$ parallel processors.*

For our main VDF construction, we assume the (σ, p)-RSW sequentiality assumption for some large parallelism function $p(\lambda, T) = \lambda^{\omega(1)}$ and sequentiality parameter $\sigma(T) = T(1 - o(1))$ to obtain a VDF with verification time 2^{λ^ϵ}. By redefining the security parameter, this leads to a VDF with poly-time verification that can evaluate up to quasi-polynomial time computation. As discussed in the introduction, other parameter settings are possible (under different hardness assumptions).

4.2.2 Learning with Errors

The following preliminaries about the Learning with Errors (LWE) problem are based on [Pei16].

Definition 4.9 (LWE Distribution)**.** *For any $\mathbf{s} \in \mathbb{Z}_q^n$ and any distribution $\chi \subseteq \mathbb{Z}_q$, the* LWE distribution *$A_{\mathbf{s},\chi} \in \mathbb{Z}_q^n \times \mathbb{Z}_q$ is sampled by choosing $\mathbf{a} \in \mathbb{Z}_q^n$ uniformly at random, sampling $e \leftarrow \chi$, and outputting $(\mathbf{a}, b = \langle \mathbf{s}, \mathbf{a} \rangle + e)$.*

Definition 4.10 (Decision LWE)**.** *Let $m = m(n) \geq 1$, $q = q(n) \geq 2$ be integers, and let $\chi(n)$ be a probability distribution on $\mathbb{Z}_{q(n)}$. The* **Decision-LWE**$_{n,m,q,\chi}$ *problem, parameterized by n, is to distinguish whether $m(n)$ independent samples are drawn from $A_{\mathbf{s},\chi}$ (for $\mathbf{s}$ that is sampled uniformly at random) or are drawn from the uniform distribution.*

For the rest of this paper, we will write LWE in place of **Decision-LWE**. Next, we consider quantitative hardness assumptions related to LWE.

Definition 4.11 ((T, δ)-LWE assumption)**.** *Any $T(n)$-time algorithm $\mathcal{A}$ solves $\mathsf{LWE}_{n,m,q,\chi}$ with distinguishing advantage at most $O(\delta(n))$.*

The discrete Gaussian distribution with mean c and standard deviation parameter s is a distribution supported over $\mathbb{Z}$ and assigns probability mass $\rho_{c,s}(x) \propto e^{-\pi(x-c)^2/s^2}$ to a number $c \in \mathbb{Z}$.

Worst-Case to Average-Case Reduction. When the LWE error distribution is instantiated with a discrete Gaussian distribution, we obtain a beautiful worst-case to average-case reduction which says that solving LWE gives us a *worst-case* algorithm for an approximate decisional version of the lattice shortest vector problem. The connection is stated formally below, with the most general version due to Brakerski et al. [BLP$^+$13].

Theorem 4.12. *[Reg05, BLP$^+$13, PRSD17] Let n, m, q, χ be parameters that define the LWE problem as above, where χ is the discrete Gaussian distribution over $\mathbb{Z}$ with parameter αq for some $\alpha = \alpha(n)$. If the $(T(n), \delta(n))$-LWE assumption is false, then there is a $T'(n)$-time algorithm for the worst-case $\widetilde{O}(n/\alpha)$-approximate GapSVP problem on n-dimensional lattices where $T' = \mathsf{poly}\Big(n, m, \log q, T, 1/\delta\Big)$.*

Moreover, the space complexity of this worst-case algorithm is bounded by $\mathsf{poly}\Big(n, m, q, T, \log(1/\delta)\Big)$.

4.2.3 Correlation Intractable Hash Families

Definition 4.13. *For a pair of efficiently computable functions $(n(\cdot), m(\cdot))$, a* hash family *with input length n and output length m is a collection $\mathcal{H} = \{h_\lambda : \{0,1\}^{s(\lambda)} \times \{0,1\}^{n(\lambda)} \to \{0,1\}^{m(\lambda)}\}_{\lambda \in \mathbb{N}}$ of keyed hash functions, along with a pair of p.p.t. algorithms:*

- $\mathcal{H}.\mathsf{Gen}(1^\lambda)$ *outputs a hash key $k \in \{0,1\}^{s(\lambda)}$.*
- $\mathcal{H}.\mathsf{Hash}(k, x)$ *computes the function $h_\lambda(k, x)$. We may use the notation $h(k, x)$ to denote hash evaluation when the hash family is clear from context.*

As in prior works [CCH$^+$19, PS19] we consider the security notion of correlation intractability [CGH98] for single-input relations and its restriction to (single-input) functions.

Definition 4.14 (Correlation Intractability)**.** *For a given relation ensemble $R = \{R_\lambda \subseteq \{0,1\}^{n(\lambda)} \times \{0,1\}^{m(\lambda)}\}$, a hash family $\mathcal{H} = \{h_\lambda : \{0,1\}^{s(\lambda)} \times \{0,1\}^{n(\lambda)} \to$*

$\{0,1\}^{m(\lambda)}\}$ *is said to be* R-correlation intractable with security (s,δ) *if for every* s*-size* $\mathcal{A}=\{\mathcal{A}_\lambda\}$,

$$\Pr_{\substack{k\leftarrow\mathcal{H}.\mathsf{Gen}(1^\lambda)\\ x\leftarrow\mathcal{A}(k)}}\Big[\big(x,h(k,x)\big)\in R\Big]=O(\delta(\lambda)).$$

We say that $\mathcal{H}$ *is* R-correlation intractable with security δ *if it is* (λ^c,δ)*-correlation intractable for all* $c>1$*. Finally, we say that* $\mathcal{H}$ *is* R-correlation intractable *if it is* $(\lambda^c,\frac{1}{\lambda^c})$*-correlation intractable for all* $c>1$*.*

To allow for a uniform security reduction in our results, we also consider the following modified definition.[14]

Definition 4.15 (Correlation Intractability against Uniform Adversaries)**.** *Let* $\mathcal{R}$ *denote a collection of relation ensembles with input length function* $n(\cdot)$ *and output length function* $m(\cdot)$*. A hash family* $\mathcal{H}$ *is said to be* $\mathcal{R}$-correlation intractable with security (T,δ) against uniform adversaries *if every* T*-time adversary* $\mathcal{A}$ *wins the following game with probability at most* $O(\delta(\lambda))$*:*

1. $\mathcal{A}(1^\lambda)$ *outputs the description of a relation* $R\in\mathcal{R}$ *and sends it to a challenger.*
2. *The challenger samples a hash key* $k\leftarrow\mathcal{H}.\mathsf{Gen}(1^\lambda)$ *and sends* k *to* $\mathcal{A}$*.*
3. $\mathcal{A}$*, given* k*, returns an input* $x\in\{0,1\}^{n(\lambda)}$*.* $\mathcal{A}$ *wins if* $(x,h_k(x))\in R$*.*

Definition 4.16 (Correlation Intractability for Functions)**.** *For a given function ensemble* $\mathcal{F}=\{f_\lambda:\{0,1\}^{n(\lambda)}\to\{0,1\}^{m(\lambda)}\}$*, a hash family* $\mathcal{H}=\{h_\lambda:\{0,1\}^{s(\lambda)}\times\{0,1\}^{n(\lambda)}\to\{0,1\}^{m(\lambda)}\}$ *is said to be* f-correlation intractable with security (s,δ) *if for every* s*-size* $\mathcal{A}=\{\mathcal{A}_\lambda\}$,

$$\Pr_{\substack{k\leftarrow\mathcal{H}.\mathsf{Gen}(1^\lambda)\\ x\leftarrow\mathcal{A}(k)}}\Big[h(k,x)=f(x)\Big]=O(\delta(\lambda)).$$

We say that $\mathcal{H}$ *is* f-correlation intractable with security δ *if it is* (λ^c,δ)*-correlation intractable for all* $c>1$*. Finally, we say that* $\mathcal{H}$ *is* f-correlation intractable *if it is* $(\lambda^c,\frac{1}{\lambda^c})$*-correlation intractable for all* $c>1$*.*

[14] This was implicit in prior works, but we make the distinction explicit here.

Remark 4.17. *We can define correlation intractability for functions against uniform adversaries similarly to Definition 4.15.*

We note that syntactically, correlation intractability for functions implies correlation intractability for relations that are implicitly described by (partial) functions.

Definition 4.18 (Unique Output Relation)**.** *We say that a relation R is a* unique output relation *if for every input x, there exists at most one output y such that $(x, y) \in R$.*

Lemma 4.19. *Suppose that $\mathcal{R}$ is a class of unique output relations. Let $\mathcal{F}$ denote a class of functions such that for all $R \in \mathcal{R}$, there exists a function $f \in \mathcal{F}$ "explaining R" in the sense that for all $(x, y) \in \{0,1\}^* \times \{0,1\}^*$, if $R(x, y) = 1$ then $f(x) = y$. Then, if a hash family $\mathcal{H}$ is correlation intractable for $\mathcal{F}$, then it is correlation intractable for $\mathcal{R}$ with the same parameters.*

In our constructions, we will make use of the correlation intractable hash family of [PS19]; in particular, we make use of the fact that it inherits strong quantitative security from the underlying LWE assumption.

Theorem 4.20 ([PS19], slightly modified)**.** *Assume the $(T \cdot n^{\omega(1)}, \delta)$-hardness of $\mathsf{LWE}_{n,m+1,q,\chi}$ for sufficiently large $q = \mathsf{poly}(n, m)$ and $m = n\lceil \log q \rceil$). Then, for every polynomial function $\ell(n)$, there is a hash family $\mathcal{H} = \{h_\lambda : \{0,1\}^s \times \{0,1\}^\ell \to \{0,1\}^m\}$ that is $(T \cdot n^{\omega(1)}, \delta)$-correlation intractable for all T-time computable functions $f : \{0,1\}^\ell \to \{0,1\}^m$.*

Proof (sketch). The [PS19] construction (making use of a polynomial modulus q) consists of two parts: a hash family for branching programs, followed by a "bootstrapping step" via levelled FHE. The security of the bootstrapping step follows from a comparatively weaker LWE security invocation (as a larger security parameter for the FHE scheme can be chosen without affecting the output length of the overall hash function), so we focus on the branching program step. Their hash function for branching programs is constructed to have output length $n\lceil \log q \rceil$ and has a security proof consisting of two steps: a "leftover hash lemma" argument for the (statistically hiding) fully homomorphic commitments, and a direct invocation of $\mathsf{LWE}_{n,m+1,q,\chi}$. By choosing large enough public parameters for the fully homomorphic commitment scheme (which does not effect the output length of the hash function), the leftover hash lemma can be made to guarantee q^{-n}-statistical indistinguishability of this step in the security proof. Finally, the security reduction from $\mathsf{LWE}_{n,m+1,q,\chi}$ runs in time $T \cdot \mathsf{poly}(n, \log q)$. This completes the proof of Theorem 4.20. $\square$

Remark 4.21. *In our later constructions, we will consider functions f computed in an "online-offline" model, where $f \in \mathsf{Size}(S)$ is computable by a size S circuit C, but the circuit requires time $T \gg S$ to construct. Theorem 4.20 above then says that correlation intractability for f can be built from a* non-uniform *LWE assumption for size $S \cdot n^{\omega(1)}$-size adversaries, but the same argument shows that one can instead rely on a* uniform *LWE assumption for time $T \cdot n^{\omega(1)}$ adversaries.*

4.2.4 Interactive Proofs and Arguments

We being by recalling the definitions of interactive proofs and arguments.

Definition 4.22. *An* interactive proof *(resp.,* interactive argument*) for a promise problem $\mathcal{L} = (\mathcal{L}_{\mathsf{yes}}, \mathcal{L}_{\mathsf{no}})$ is a pair (P, V) of interactive algorithms satisfying:*

- **Completeness.** *For any $x \in \mathcal{L}_{\mathsf{yes}}$, when P and V interact on common input x, the verifier V outputs* 1 *with probability* 1.

- **Soundness.** *For any $x \in \mathcal{L}_{\mathsf{no}} \cap \{0,1\}^n$ and any* unbounded *(resp.,* polynomial-time*) interactive P^*, when P^* and $V(x)$ interact, the probability that V outputs* 1 *is a negligible function of n.*

The protocol is public coin *if each of V's messages is an independent uniformly random string of some length (and the verifier's decision to accept or reject does not use any secret state). In this setting, we will denote prover messages by $(\alpha_1, \ldots, \alpha_\ell)$ and verifier messages by $(\beta_1, \ldots, \beta_{\ell-1})$ in a $2\ell - 1$-round protocol.*

Definition 4.23. *A* non-interactive *argument scheme (in the CRS model) is for a promise problem* $\mathcal{L} = (\mathcal{L}_{\mathsf{yes}}, \mathcal{L}_{\mathsf{no}})$ *is a triple* (Setup, P, V) *of* non-interactive *algorithms with the following properties:*

- $\mathsf{Setup}(1^n)$ *outputs a common reference string* crs.
- $P(\mathsf{crs}, x)$ *outputs a proof* π.
- $V(\mathsf{crs}, x, \pi)$ *outputs a bit* $b \in \{0, 1\}$

It satisfies the notions of completeness and (computational) soundness as above.

Remark 4.24. *Given an argument system* Π*, we consider three important complexity measures of* Π*:*

- *The runtime of the prover* P *on an instance of size* n*.*
- *The quantitative soundness of* Π*; that is, how long a cheating prover* P^* *can run with the guarantee that soundness is unbroken.*
- *The runtime of the verifier* V *on an instance of size* n*. For a nontrivial argument system, this quantity should be smaller than the previous two.*

In this paper, we will sometimes consider non-interactive protocols with a crs *whose length is superpolynomial in the instance size* n *or security parameter* λ*. In this situation, we will still parameterize prover efficiency, verifier efficiency, and quantative soundness as functions of* (n, λ) *rather than the Prover/Verifier input length (which is at least the length of the* crs*).*

Definition 4.25 (Fiat-Shamir Transform)**.** *Let* Π *denote a public coin interactive proof (or argument) system* Π *that has* ℓ *prover messages and* $\ell - 1$ *verifier messages of length* $m = m(\lambda)$*. Then, for a hash family* $\mathcal{H} = \{\{h_k : \{0,1\}^* \to \{0,1\}^{m(\lambda)}\}_{k \in \{0,1\}^\lambda}\}_\lambda$*, we define the Fiat-Shamir non-interactive protocol* $\Pi_{\mathrm{FS},\mathcal{H}} = (\mathsf{Setup}, P_{\mathrm{FS}}, V_{\mathrm{FS}})$ *as follows:*

- $\mathsf{Setup}(1^\lambda)$*: sample a hash key* $k \leftarrow \mathcal{H}.\mathsf{Gen}(1^\lambda)$*.*

- *$P_{\text{FS}}(x)$: for $i \in \{1, \ldots, \ell\}$, recursively compute the following pairs (α_i, β_r):*

  - *Compute $\alpha_i = P(\tau_i$ for $\tau_i = (x, \alpha_1, \beta_1, \ldots, \alpha_{i-1}, \beta_{i-1})$.*

  - *Compute $\beta_i = h_k(\tau_{i-1}, \alpha_i)$.*

  Then, $P_{\text{FS}}(x)$ outputs $\pi = (\alpha_1, \beta_1, \ldots, \alpha_\ell)$.

- *$V_{\text{FS}}(\mathsf{crs}, x, \pi)$ parses $\pi = (\alpha_1, \beta_1, \ldots, \alpha_\ell)$ and verifies that:*

  - *$\beta_i = h_k(\tau_{i-1}, \alpha_i)$ for all $1 \le i \le \ell - 1$, and*

  - *$V(x, \pi) = 1$.*

We note the following facts about $\Pi_{\text{FS},\mathcal{H}}$

- *The honest prover complexity of $\Pi_{\text{FS},\mathcal{H}}$ is equal to the honest prover complexity of Π with an additive overhead of computing $\ell - 1$ hash values.*

- *The verifier complexity of $\Pi_{\text{FS},\mathcal{H}}$ is equal to the verifier complexity of Π with the same hashing additive overhead.*

- *The protocol $\Pi_{\text{FS},\mathcal{H}}$ is not necessarily sound, even if Π is sound and $\mathcal{H}$ is a "strong cryptographic hash function."*

Finally, we define the notion of *unambiguous soundness* [RRR16], which is crucial for our PPAD-hardness result. For non-interactive arguments, the soundness notion we consider is *adaptive* in that we allow the prover P^* to adaptively choose the statement x after seeing the crs.

Definition 4.26 (Unambiguous Soundness [RRR16, CHK+19a])**.** *A public-coin interactive proof system Π is* unambiguously sound *if (1) it is sound, and (2) for every $x \in L$ and every (complete) collection of verifier messages $(\beta_1, \ldots, \beta_{\ell-1})$, there exists a distinguished proof $\pi^*(x, \beta_1, \ldots, \beta_{\ell-1})$ such that the following soundness condition holds: For all $x \in L$ and all cheating provers P^*, the probability that the transcript $\langle P^*(x), V(x) \rangle$ contains a proof π such that $V(x, \pi) = 1$ and $\pi \neq \pi^*(x, \beta_1, \ldots, \beta_{\ell-1})$ is negligible.*

Definition 4.27 (Adaptive Unambiguous Soundness)**.** *A non-interactive argument system* $\Pi = (\mathsf{Setup}, P, V)$ *is* adaptively unambiguously sound *against (uniform or nonuniform) time T adversaries if for all instances $x \in L$ and all common reference strings* crs*, there exists a "distinguished proof"* $\pi^*(\mathsf{crs}, x)$ *such that the following soundness condition holds: For all time T cheating provers P^*, the probability that* $P^*(\mathsf{crs}) = (x, \pi)$ *where* $V(x, \pi) = 1$ *and* either $x \notin L$ *or* $\pi \neq \pi^*(\mathsf{crs}, x)$ *is negligible.*

4.2.5 Non-trivial Preprocessing Algorithms for the Discrete Logarithm Problem

In this section, we describe a family of randomized algorithms for solving the (worst-case) discrete logarithm problem over $\mathbb{Z}_p^\times$ for a prime p. This will be necessary for the analysis of our variant of Pietrzak's interactive proof system of repeated squaring, and for its associated Fiat-Shamir hash function.

The algorithm is a simple variant of the index calculus algorithm, as presented in [CCRR18], but with different parameter choices. We present the algorithm, analyze its runtime, and state (with citation) its success probability.

Given an arbitrary generator g for $\mathbb{Z}_p^\times$ for $p = 2^{O(\lambda)}$ and a time bound t, we consider the following *preprocessing algorithm* for discrete logarithms with base g.

- **Offline Phase**: for all $1 \leq k \leq t$, compute the discrete logarithm of k in base g, and store the answer α_k.

- **Online Phase**: given challenge h, define $h' = h \cdot g^{-r}$ for a uniformly random r, and check if $h' \in \mathbb{Z}$ factors into a product of elements of the set $\{2, \ldots, t\}$. If such a factorization $h' = k_1 \cdot \ldots \cdot k_\ell$ is found, then output the discrete logarithm $r + k_1 + k_2 + \ldots + k_\ell$. Otherwise, output $\perp$.

For a runtime analysis, note that each discrete logarithm in the offline phase can be computed in time $2^{\tilde{O}(\lambda^{1/2})}$ via the algorithm of [Adl79, Pom87], so the entire offline phase can be computed in time $t \cdot 2^{\tilde{O}(\lambda^{1/2})}$.

The online phase can be computed in time $t \cdot \mathsf{poly}(\lambda)$, with the most expensive step being the attempted factorization of h' via trial division.

Finally, since h' is a uniformly random element of $\{1, \ldots, p-1\}$, the success probability of one iteration of the online phase is simply the probability that a random element of $\{1, \ldots, p-1\}$ has no prime factor larger than t. Based on smooth number estimates (such as those following from [CEP83]; see [Gra08] for a survey of results), we note the following special cases.

Theorem 4.28 (Follows from [CEP83])**.** *The following probability bounds hold:*

- *For $t = \lambda^A$, the algorithm has success probability at least $2^{\frac{-\lambda}{A}(1-o(1))}$.*
- *For $t = 2^{\log(\lambda)^c}$, the algorithm has success probability at least $2^{\frac{-\lambda}{\log(\lambda)^{c-1}}(1-o(1))}$.*
- *For sufficiently large $t = 2^{\tilde{O}(\lambda^\epsilon)}$, the algorithm has success probability at least $2^{\frac{-\lambda^{1-\epsilon}(1-o(1))}{\log^2(\lambda)}}$.*

We note that this algorithm, in the regime $t = \mathsf{poly}(\lambda)$, was considered in [CCRR18] as evidence against the optimal security of discrete log over $\mathbb{Z}_p^\times$; a simple application of Rankin's method [Ran38] sufficed for their calculations, but we are interested in analyzing larger values of t.

4.3 Correlation Intractability for Special Inefficient Functions

In this section, we show how to construct correlation-intractable hash families that support certain functions f that are *not necessarily efficiently computable.* Specifically, we handle functions that can be computed by a randomized algorithm that is only correct with low probability(Section 4.3.1). As a special case (by appealing to Section 4.2.5), this implies that we can handle functions f that are efficient given a small number of calls to a discrete log oracle (Section 4.3.2).

4.3.1 A Self-Reduction for Correlation Intractability

We first show the following simple self-reduction for correlation-intractable hash families.

Theorem 4.29. *If a hash family $\mathcal{H}$ is (s, δ)-correlation intractable for all non-uniform time t-computable functions, then it is $(s, \frac{\delta}{\epsilon})$-correlation intractable for all functions f that are computable in the following preprocessing model:*

- ***Preprocessing Phase:*** *In unbounded time, output the description of a randomized function g_r running in time t.*
- ***Online Phase:*** *Given an input x, compute $g_r(x)$.*
- ***Correctness Guarantee:*** *For all inputs x, we have that $\Pr[g_r(x) = f(x)] \geq \epsilon$.*

Proof. Given a function f computable in the above preprocessing model, suppose that an adversary $\mathcal{A}$ breaks the $(s, \frac{\delta}{\epsilon})$-correlation intractability of $\mathcal{H}$. Then, $\mathcal{A}(k)$ finds an input x such that $h_k(x) = f(x)$ with probability at least $\frac{\delta}{\epsilon}$. But for a uniformly random r, we are guaranteed that (for any fixed x), $f(x) = g_r(x)$ with probability at least ϵ. From this, we conclude that for a random r, the exact same adversary $\mathcal{A}(k)$ finds an input x such that $h_k(x) = g_r(x)$ with probability at least δ, breaking the (s, δ)-correlation intractability of $\mathcal{H}$. □

Remark 4.30. *If the preprocessing phase of this online-offline algorithm can be implemented in some (uniform) time T, then correlation intractability against uniform adversaries (with the appropriate parameters) is also preserved.*

4.3.2 CI for Efficient Functions Relative to Discrete-Log

By combining Theorem 4.29 with the non-trivial discrete log algorithms in Section 4.2.5 as well as the construction of correlation-intractable hash families due to [PS19] (Theorem 4.20), we obtain a CI construction for a class of functions that are efficient relative to a discrete log oracle. We formalize the result as follows.

Definition 4.31. *We say that a function f is (T, q, ℓ)-*computable given a discrete log oracle *if f is computable by an oracle algorithm $A^{\mathcal{O}(\cdot)}$, where*

- *A runs in time T,*
- *A makes at most q queries to $\mathcal{O}$,*
- *Every query (g, h, p) to $\mathcal{O}$ has length at most ℓ, and*
- *$\mathcal{O}(g, h, p)$ computes the discrete logarithm of h with respect to g in the group $\mathbb{Z}_p^\times$.*

Theorem 4.32. *Let $\epsilon > 0$ be arbitrary. Assume that (decision)* LWE *is $\left(2^{\tilde{O}(n^{1/2})}, 2^{-n^{1-\epsilon}}\right)$-hard (or alternatively, $\left(2^{\tilde{O}(n^{\epsilon})}, 2^{-n^{1-\epsilon}}\right)$-hard for non-uniform algorithms) for some $q = \mathsf{poly}(n)$. Then, for $m = n \log q$ and every polynomial function $\ell(n)$, there exists a hash family $\mathcal{H}$ mapping $\{0,1\}^{\ell(n)} \to \{0,1\}^m$ such that*

- *$\mathcal{H}$ is correlation intractable for all functions f that are $(2^{n^\epsilon}, \mathsf{poly}\log n, \tilde{O}(m))$-computable given a discrete log oracle, and*
- *A hash function h from $\mathcal{H}$ can be evaluated in time $2^{\tilde{O}(n^\epsilon)}$.*

Under the assumption that (decision) LWE *is $\left(2^{\tilde{O}(n^{1/2})}, 2^{-\frac{n}{\log^c n}}\right)$-hard for some constant $c > 0$ (or alternatively, $\left(\mathsf{quasipoly}(n), 2^{-\frac{n}{\log^c n}}\right)$-hard for non-uniform algorithms), there exists such a hash family $\mathcal{H}$ where*

- *$\mathcal{H}$ is correlation intractable for all functions f that are $(\mathsf{quasipoly}(n), \mathsf{poly}\log n, \tilde{O}(m))$-computable given a discrete log oracle, and*
- *A hash function h from $\mathcal{H}$ can be evaluated in time* $\mathsf{quasipoly}(n)$.

Finally, under the assumption that (decision) LWE *is $\left(\mathsf{poly}(n), q^{-\delta n}\right)$-hard for non-uniform distinguishers (or $\left(2^{\tilde{O}(n^{1/2})}, q^{-\delta n}\right)$-hard for uniform distinguishers) for a fixed $\delta > 0$, there exists such a hash family $\mathcal{H}$ where*

- *$\mathcal{H}$ is correlation intractable for all functions f that are $(n^{1/\delta}, O(1), O(m))$-computable given a discrete log oracle, and*

- *A hash function h from $\mathcal{H}$ can be evaluated in time $n^{O(1/\delta)}$.*

Remark 4.33. *Looking ahead, Theorem 4.32 is not directly used in this work to obtain our main theorem (Theorem 4.1). The reason for this is due to technicalities about preprocessing and non-uniformity when describing the [Pie18] protocol and its bad challenge function. A more complicated version of Theorem 4.32 could be directly used to prove Theorem 4.1, but we prefer to state a simpler version of Theorem 4.32 and then directly analyze the [Pie18] protocol in Section 4.5.1.*

4.4 Round-by-Round (Unambiguous) Soundness and Fiat-Shamir

Following [CCH+18, CCH+19], we consider the notion of round-by-round soundness to capture a particular kind of soundness analysis for super-constant round interactive proofs. Since we are interested in *unambiguous* soundness for our protocol, we define an analogous notion of "unambiguous round-by-round soundness" and note (as in [CCH+18]) that correlation intractability for an appropriate relation suffices for a hash family to instantiate the Fiat-Shamir heuristic for unambiguously round-by-round sound interactive proofs.

Definition 4.34 (Unambiguous Round-by-Round Soundness, adapted from [CCH+18])**.** *Let $\Pi = (P, V)$ be a $2\ell - 1$-message public coin interactive proof system for a language L.*

We say that Π has unambiguous round-by-round soundness error *$\epsilon(\cdot)$ if there exist functions* $(\mathsf{State}, \mathsf{NextMsg})$ *with the following syntax.*

- State *is a deterministic (not necessarily efficiently computable) function that takes as input an instance x and a transcript prefix τ and outputs either* acc *or* rej.

- $\mathsf{NextMsg}$ *is a deterministic (not necessarily efficiently computable) function that takes as input an instance x and a transcript prefix τ and outputs a (possibly* *aborting) prover message $\alpha \in \{0,1\}^* \cup \{\perp\}$.*

We additionally require that the following properties hold.

1. *If $x \notin L$, then $\mathsf{State}(x, \emptyset) = \mathsf{rej}$, where $\emptyset$ denotes the empty transcript.*

2. *If $\mathsf{State}(x, \tau) = \mathsf{rej}$ for a transcript prefix τ, then $\mathsf{NextMsg}(x, \tau) = \perp$. That is, $\mathsf{NextMsg}(x, \tau)$ is only defined on accepting states.*

3. *For every input x and partial transcript $\tau = \tau_i$, then for every potential prover message $\alpha_{i+1} \neq \mathsf{NextMsg}(x, \tau)$, it holds that*

$$\Pr_{\beta_{i+1}}\left[\mathsf{State}\Big(x, \tau|\alpha_{i+1}|\beta_{i+1}\Big) = \mathsf{acc}\right] \leq \epsilon(n)$$

4. *For any* full[15] *transcript τ, if $\mathsf{State}(x, \tau) = \mathsf{rej}$ then $V(x, \tau) = 0$.*

We say that Π is unambiguously round-by-round sound *if it has unambiguous round-by-round soundness error ϵ for some $\epsilon(n) = \mathrm{negl}(n)$.*

Remark 4.35. *Note that a proof system that satisfies unambiguous round-by-round soundness also satisfies standard unambiguous soundness. Indeed, if a proof system Π satisfies unambiguous round-by-round soundness, every statement $x \in L$ and collection of verifier messages $(\beta_1, \ldots, \beta_{\ell-1})$ has an associated "distinguished proof" defined by iterating the* NextMsg *function on the appropriate partial transcripts. It is (statistically) hard for a cheating prover P^* to find any proof $\tilde{\pi}$ other than $\pi^* = \pi^*(x, \beta_1, \ldots, \beta_{\ell-1})$ because finding such a proof violates unambiguous round-by-round soundness at whichever round $\tilde{\pi}$ first deviates from π^*.*

With this definitional framework, a direct adaptation of ([CCH+18], Theorem 5.8) yields the following result.

Theorem 4.36. *Suppose that $\Pi = (P, V)$ is a $2\ell - 1$-message public-coin interactive proof for a language L with perfect completeness and unambiguous round-by-round soundness with corresponding functions $(\mathsf{State}, \mathsf{NextMsg})$. Let X_n denote the set of*

[15] By a full transcript, we mean a transcript for which the verifier halts.

partial transcripts (including the input and all messages sent) and let Y_n denote the set of verifier messages when Π is executed on an input of length n.

Finally, define the relation ensemble $R = R_{\mathsf{State},\mathsf{NextMsg}}$ as follows:

$$R^{(n)}_{\mathsf{State},\mathsf{NextMsg}} \overset{\mathsf{def}}{=} \left\{ \Big(\big(x, \tau | \alpha \big), \beta \Big) : \begin{array}{c} x \in \{0,1\}^n, \\ \alpha \neq \mathsf{NextMsg}(x, \tau) \\ \textit{and} \\ \mathsf{State}(x, \tau|\alpha|\beta) = \mathsf{acc} \end{array} \right\}.$$

If a hash family $\mathcal{H} = \{\mathcal{H}_n : X_n \to Y_n\}$ is $T \cdot \lambda^{\omega(1)}$-correlation intractable for R, then the round-reduced protocol $\Pi_{\mathrm{FS},\mathcal{H}}$ is an adaptively unambiguously sound argument system (against time $T \cdot \lambda^{\omega(1)}$ cheating provers) for L.

Finally, we consider the special case where the relation $R_{\mathsf{State},\mathsf{NextMsg}}$ associated to a protocol Π is a unique output relation (Definition 4.18).

Definition 4.37 (Bad Challenge Function)**.** *Let Π denote a public-coin interactive proof system satisfying unambiguous round-by-round soundness with associated functions* $(\mathsf{State}, \mathsf{NextMsg})$. *Suppose that the relation $R_{\mathsf{State},\mathsf{NextMsg}}$ as defined above is a unique output relation.*

We say that a function $f_{\mathsf{State},\mathsf{NextMsg}}$ is a bad challenge function *for Π if for all partial transcripts (x, τ), and all verifier messages β, if $(x|\tau, \beta) \in R_{\mathsf{State},\mathsf{NextMsg}}$, then $\beta = f_{\mathsf{State},\mathsf{NextMsg}}(x, \tau)$*

Invoking Lemma 4.19 and Theorem 4.36, we obtain the following corollary.

Corollary 4.38. *In the setting of Theorem 4.36, if f is a bad challenge function for Π and $\mathcal{H}$ is $T \cdot \lambda^{\omega(1)}$-correlation intractable for f, then $\Pi_{\mathrm{FS},\mathcal{H}}$ is an adaptively unambiguously sound non-interactive argument system against $T \cdot \lambda^{\omega(1)}$-time cheating provers.*

4.5 Fiat-Shamir for the Repeated Squaring Protocol

In this section, we describe our variant of the [Pie18] repeated squaring protocol, analyze its round-by-round unambiguous soundness (Definition 4.34), and show that the protocol has an associated bad-challenge function (Definition 4.37) that allows for the desired Fiat-Shamir instantiation (Theorem 4.1).

4.5.1 Our Variant of the Repeated Squaring Protocol

For ease of notation and analysis, we adopt the following variant of Pietrzak's protocol [Pie18]. While it is essential for us to use a protocol with unambiguous soundness, our deviation from the variant of [CHK+19b] is voluntary. For simplicity, we only consider $T = 2^t$ to be a power of 2.

- **Setup**: Sample $(N, g) \leftarrow \mathsf{Setup}(1^\lambda)$ for the RSW function (Section 4.2.1).
- **Initial Claim:** On input T, the prover outputs $h = g^{2^T} = f_{N,g}(T)$. The implicit claim is that h is indeed equal to $f_{N,g}(T)$.
- **Round-by-Round Reduction** given a claim (N, T, g_i, h_i), the prover and verifier execute a 2-round reduction step that outputs a new claim:
    - With $\frac{T}{2} + O(1)$ group operations, the prover computes $u_i = g_i^{2^{\frac{T}{2}}}$ along with the unique square root v_i of u_i such that $v_i \in \mathbb{QR}_N^+$. In particular, this v_i is equal to one of $\pm g_i^{2^{T/2-1}}$. The prover outputs (u_i, v_i).
    - The verifier checks that $v_i \in \mathbb{QR}_N^+$ and that $v_i^2 = u_i$; if a check fails, the verifier aborts. Otherwise, the verifier samples a random string $r_i \leftarrow \{0,1\}^\lambda$.
    - The prover and verifier recurse on the new claim $(N, T/2, g_{i+1} = u_i \cdot g_i^r, h_{i+1} = h_i \cdot u_i^r)$.

- **Base Case**: On the final claim $(N, 1, g_t, h_t)$, the verifier accepts if and only if $h_t = g_t^2$.

We denote this main interactive protocol by Π. We now proceed to analyze its soundness properties.

4.5.2 Unambiguous Round-by-Round Soundness and Bad-Challenge Function

We show that Π satisfies unambiguous round-by-round soundness and has an associated bad challenge function $f : \mathbb{Z}_N \times \mathbb{Z}_N \to \{0,1\}^\lambda$ that has a non-trivial preprocessing algorithm.

We begin by defining the functions $(\mathsf{State}, \mathsf{NextMsg})$, using the fact that every partial transcript (x, τ) has an associated "current claim".

- $\mathsf{State}(x, \tau)$ is defined to be acc if and only if all prover messages (u, v) pass the verifier's local check (that $v^2 = u$ and $v \in \mathbb{QR}_N^+$) and the "current claim" of the form $h_i = g_i^{T_i}$ is true.
- $\mathsf{NextMsg}(x, \tau)$ is defined (for accepting states) to be (u_i, v_i) for $u_i = g_i^{T_i/2}$ and $v_i \in \pm g_i^{T_i/2-1}$ the appropriately chosen square root in $\mathbb{QR}_N^+$. For rejecting states, $\mathsf{NextMsg}(x, \tau) = \bot$ by definition.

Theorem 4.39. *The protocol* Π *satisfies unambiguous round-by-round soundness with associated functions* $(\mathsf{State}, \mathsf{NextMsg})$*. Moreover,* Π *has a bad challenge function* f*.*

Proof. Properties (1), (2), and (4) of unambiguous round-by-round soundness follow immediately from the definitions of $(\mathsf{State}, \mathsf{NextMsg})$. What remains is to verify property (3), which follows from two facts that we will prove:

- At each step i of the round-by-round reduction, if Claim i is false, then for every prover message (u_i, v_i), there is at most one challenge r^* such that Claim $i+1$ is true.

- At each step i, if Claim i is true, then for every prover message (u_i, v_i) *that deviates* from the correct messages, there is at most one challenge r^* such that Claim $i+1$ is true.

To prove this, we consider the reduction step for an arbitrary verifier message r_i:

$$h_{i+1} := h_i \cdot u_i^r, g_{i+1} = u_i \cdot g_i^r$$

Let (η, ω, γ) denote the discrete logarithms of (h_i, g_i, u_i), respectively, in base g. We then see that Claim $i+1$ is true if and only if

$$\eta + r \cdot \omega \equiv 2^{T_i/2}(\omega + r \cdot \gamma) \pmod{p'q'},$$

which is true if and only if

$$r(\omega - 2^{T_i/2}\gamma) \equiv 2^{T_i/2}\omega - \eta \pmod{p'q'}.$$

We then have two cases to analyze:

- **Case 1:** If $\omega = 2^{T_i/2}\gamma$, then the equality above holds if and only if $\eta = 2^{T_i/2}\omega$ as well, which is exactly the case that Claim i was true and u_i is the correct prover message. v_i must additionally be the correct prover message because of the verifier's local check.
- **Case 2:** If $\omega \neq 2^{T_i/2}\gamma$, then either the verifier rejects some pair (u, v) (if the local check on (u, v) fails) or we are guaranteed that $\omega - 2^{T_i/2}\gamma \not\in \{0, p'q'\}$ (because we are guaranteed that g_i and u_i are both in $\mathbb{QR}_N$). This implies that $\omega - 2^{T_i/2}$ has additive order at least $\min(p', q')$, and hence there is at most one choice of r satisfying the above equation in the range $\{0, 1, \ldots, 2^\lambda - 1\}$.

This completes the analysis. In fact, the analysis above shows that for every step of the round-by-round reduction, there is a bad challenge function $f_i(N, g, T_i, g_i, h_i, u_i)$ governing the soundness of the ith reduction, so we also conclude the existence of a bad challenge function f. □

Having showed that Π has a bad challenge function f, we now describe and analyze an algorithm for computing it.

First, we note that the function f_i can be computed exactly as follows:

1. Given $(N, g, T_i, g_i, h_i, u_i)$, compute the three discrete logarithms η, ω, γ as above as well as the factorization $N = pq = (2p' + 1)(2q' + 1)$.

2. Solve the linear equation

$$r(\omega - 2^{T_i/2}\gamma) \equiv 2^{T_i/2}\omega - \eta \pmod{p'q'}.$$

   for r, and output the unique solution r^* (if one exists) in the range $[2^\lambda]$. This second step is efficient: first compute $2^{T_i/2} \pmod{p'q'}$, and then solve the linear equation via a GCD computation.

Since step (1) of this computation is extremely inefficient to compute exactly, this description is insufficient for our purposes. However, by invoking Theorem 4.28, we can show the following efficiency property of f.

Theorem 4.40. *The bad challenge function f can be computed by a preprocessing algorithm with any one of the three following efficiency guarantees:*

- *Offline time $2^{\tilde{O}(\lambda^{1/2})}$, online time $2^{\tilde{O}(\lambda^{\epsilon})}$, and success probability $2^{-\Omega\left(\frac{\lambda^{1-\epsilon}}{\log^2(\lambda)}\right)}$*

- *Offline time $2^{\tilde{O}(\lambda^{1/2})}$, online time $2^{\log(\lambda)^c}\cdot\mathsf{poly}(\lambda)$, and success probability $2^{\frac{-\Omega(\lambda)}{\log(\lambda)^{c-1}}(\frac{1}{6}-o(1))}$*

- *Offline time $2^{\tilde{O}(\lambda^{1/2})}$, online time $\lambda^{1/\delta}\cdot\mathsf{poly}(\lambda)$, and success probability $2^{-\delta\lambda(\frac{1}{6}-o(1))}$.*

Proof. The algorithm is as follows.

- Offline phase: factor N in time $2^{\tilde{O}(\lambda^{1/2})}$ using Dixon's factorization method [Dix81, Pom87]. Also, compute $\tilde{g}$, a square root of g that has order $\phi(N)/2$.

- Compute the discrete logarithms of g_i, h_i, u_i (in base $\tilde{g}$) modulo p and the discrete logarithms of g_i, h_i, u_i (in base $\tilde{g}$) modulo q using the preprocessing algorithm from Section 4.2.5.[16] With the appropriate parameter choice, this contributes $2^{\tilde{O}(\lambda^{1/2})}$ offline time, $2^{\tilde{O}(\lambda^{\epsilon})}$ online time, and has success probability $2^{\frac{-\lambda^{1-\epsilon}}{\log^2(\lambda)}}$.

- Compute η, ω, and γ by halving the six discrete logarithms above and using the Chinese remainder theorem.

- Finish the computation of r^* as above.

The claimed efficiency follows directly from Theorem 4.28. □

Remark 4.41. *In order to match the preprocessing model defined in Section 4.3.1, we note that the modulus $N = pq$ is not considered part of the "input" to the protocol, but is instead considered a global public parameter.*

Finally, by combining Theorem 4.39 (the existence of a bad-challenge function f for Π), Theorem 4.40 (the low-probability preprocessing algorithm for f), Corollary 4.38 (hash families that are correlation intractable for a function f suffice to compile interactive protocols with bad-challenge function f), Theorem 4.29 (relating CI for efficient deterministic functions to CI for functions computable via low-probability preprocessing algorithms), and Theorem 4.20 (CI for efficient functions exist under LWE), we obtain Theorem 4.1, which we restate here for convenience. We note that the LWE security parameter n is related to the repeated squaring security parameter λ via the relation $\lambda = n\log(q) = O(n\log n)$.

Theorem 4.42 (Theorems 4.1 and 4.3, restated)**.** *Let $\epsilon > 0$ be arbitrary. Assume that (decision) LWE is $\left(2^{\tilde{O}(n^{1/2})}, 2^{-n^{1-\epsilon}}\right)$-hard (or alternatively, $\left(2^{\tilde{O}(n^{\epsilon})},\ 2^{-n^{1-\epsilon}}\right)$-hard for non-uniform algorithms). Then, there exists a hash family $\mathcal{H}$ that soundly instantiates the Fiat-Shamir heuristic for the [Pie18] interactive proof system. A hash function h from the family $\mathcal{H}$ can be evaluated in time $2^{\tilde{O}(\lambda^{\epsilon})}$ for repeated squaring over groups of size $2^{O(\lambda)}$ with $\lambda = O(n\log n)$.*

[16] Note that $\tilde{g}$ generates $\mathbb{Z}_p^{\times}$ and $\mathbb{Z}_q^{\times}$ when reduced modulo p and q respectively, so the hypotheses of the algorithm are satisfied.

Under the assumption that (decision) LWE *is* $\left(2^{\tilde{O}(n^{1/2})}, 2^{-\frac{n}{\log^c n}}\right)$*-hard for some constant* $c > 0$ *(or alternatively,* $\left(\mathsf{quasipoly}(n), 2^{-\frac{n}{\log^c n}}\right)$*-hard for non-uniform algorithms), there exists such a hash family* $\mathcal{H}$ *with quasi-polynomial evaluation time.*

Finally, under the assumption that (decision) LWE *is* $\left(\mathsf{poly}(n), q^{-\delta n}\right)$*-hard for non-uniform distinguishers (or* $\left(2^{\tilde{O}(n^{1/2})}, q^{-\delta n}\right)$*-hard for uniform distinguishers) for a fixed* $\delta > 0$*, there exists such a hash family* $\mathcal{H}$ *with evaluation time* $\lambda^{O(1/\delta)}$*.*

4.6 Applications to PPAD-Hardness and VDFs

Having proved Theorem 4.1, we now conclude our main applications, Theorem 4.5 and Theorem 4.6. Theorem 4.5 follows directly from Theorems 4.1 and 4.3 along with the work of [CHK$^+$19b, EFKP19, CHK$^+$19a], while Theorem 4.6 follows from Theorems 4.1 and 4.3 as an instantiation of the [Pie18] protocol in the standard model.

For each of the two applications, we state the relevant definitions and re-state the main theorems.

4.6.1 Hardness in PPAD and CLS

The following preliminaries are taken from [CHK$^+$19b]. We first recall the definition of **PPAD**.

Definition 4.43 (End-of-Line Problem)**.** *An instance of the End-of-Line (search) problem consists of a pair* (S, P) *of circuits computing functions from* $\{0,1\}^m \to \{0,1\}^m$*. We assume without loss of generality that* $P(0^m) = 0^m$ *and* $S(0^m) \neq 0^m$ *(as this can be checked efficiently). A solution to the search problem is a vertex* $v \in \{0,1\}^m$ *such that* $\mathsf{P}(\mathsf{S}(v)) \neq v$ *or* $\mathsf{S}((P(v)) \neq v \neq 0^m$*.*

Definition 4.44 (**PPAD**)**.** *The complexity class* **PPAD** *is the subclass of* **TFNP** *(search problems with efficient verification such that every instance is guaranteed to have a solution) consisting of all problems that are polynomial-time reducible to End-of-Line.*

To obtain hardness for **PPAD** (and indeed the subclass **CLS** [DP11]), we construct a hard instance of the "relaxed sink-of-verifiable-line problem" [CHK+19a].

Definition 4.45 (rSVL)**.** *An instance of the relaxed sink-of-verifiable-line (rSVL) (promise) problem consists of two circuits* (S, V)*, a distance* $L \in [2^m]$*, and a "source vertex"* $v_0 \in \{0,1\}^m$*. We are promised that for every pair* $(v, i) \in \{0,1\}^m \times [L]$ *such that* $v = \mathsf{S}^i(v_0)$*, it holds that* $\mathsf{V}(v, i) = 1$*. A solution to the problem is one of the following two types:*

- ***The sink:*** *a vertex* $v \in 0, 1^m$ *such that* $\mathsf{V}(v, L) = 1$*, or*
- ***False positive:*** *a pair* $(v, i) \in \{0,1\}^m \times [L]$ *such that* $v \neq \mathsf{S}^i(v_0)$ *but* $\mathsf{V}(v, i) = 1$*.*

We note that rSVL is itself not a total search problem, but it is known [CHK+19a] that rSVL *reduces* to some total search problems (indeed, even problems in **CLS**).

Our **CLS**-hardness result relies on the following theorem implicit in [CHK+19b].

Theorem 4.46 (Implicit in [CHK+19b])**.** *Suppose that Fiat-Shamir for the [Pie18] interactive proof system (as defined in Section 4.5.1) can be instantiated using some efficiently computable hash family* $\mathcal{H}$ *so that the resulting non-interactive argument system is adaptively unambiguously sound (Definition 4.27). Then, there is an efficient construction of a hard-on-average rSVL problem.*

We note two differences between our setting and the setting of [CHK+19b]. First, our variant of the [Pie18] is not identical to theirs; however, the differences are insubstantial to their hardness reduction.[17] Second, the verification procedure in (one variant of) our non-interactive protocol takes time $2^{\tilde{O}(\lambda^\epsilon)}$ rather than $\mathsf{poly}(\lambda)$; this is resolved by redefining the security parameter $\kappa = 2^{\tilde{O}(\lambda^\epsilon)}$ and then running their reduction to produce rSVL instances where the circuits (S, V) are $\mathsf{poly}(\kappa)$-size and the problem is hard for $\mathsf{poly}(\kappa)$-time algorithms.

With the two modifications above, by combining Theorem 4.46 with Theorems 4.1 and 4.3, we obtain our main **PPAD**-hardness result, Theorem 4.5.

[17] What is important is that our protocol satisfies adaptive unambiguous soundness and has a similarly efficient merging procedure (see Section 4.4, Property 3 in [CHK+19b]. This allows for their construction of "unambiguously sound incrementally verifiable computation" [CHK+19a] to go through.

Theorem 4.47 (Theorem 4.5, restated)**.** *For a constant $\epsilon > 0$, suppose that*

- *n-dimensional* LWE *(with polynomial modulus) is* $\left(2^{\tilde{O}(n^{1/2})}, 2^{-n^{1-\epsilon}}\right)$*-hard, and*

- *The repeated squaring problem on an instance of size 2^λ requires $2^{\lambda^\epsilon \log(\lambda)^{\omega(1)}}$ time.*

Then, there is a hard-on-average problem in $\mathbf{CLS} \subseteq \mathbf{PPAD}$*. The same conclusion holds if for some $c > 0$,*

- LWE *is* $\left(2^{\tilde{O}(n^{1/2})}, 2^{-\frac{n}{\log(n)^c}}\right)$*-hard, and*

- *The repeated squaring problem is hard for quasi-polynomial time algorithms.*

The same conclusion also holds if for some $\delta > 0$,

- LWE *is* $\left(\mathsf{poly}(n), q^{-\delta n}\right)$*-hard for non-uniform distinguishers, and*

- *The repeated squaring problem is hard for polynomial time algorithms.*

4.6.2 Verifiable Delay Functions

The following definition is taken from [BBBF18].

Definition 4.48 (Verifiable Delay Function)**.** *A verifiable delay function (VDF) is a triple of algorithms* $(\mathsf{Setup}, \mathsf{Eval}, \mathsf{Verify})$ *with the following syntax.*

- $\mathsf{Setup}(1^\lambda, t)$ *is a randomized algorithm that takes as input the security parameter 1^λ along with a time bound t. It outputs public parameters* pp.

- $\mathsf{Eval}(\mathsf{pp}, x)$ *takes an input x (along with the public parameters* pp*) and returns an output y along with a proof π.*

- $\mathsf{Verify}(\mathsf{pp}, x, y, \pi)$ *takes as input the public parameters* pp*, and input x, an output y, and a proof π. It outputs a bit $b \in \{0, 1\}$.*

The scheme must satisfy the following properties.

- ***Correctness:*** *For all* pp *in the support of the distribution* $\mathsf{Setup}(1^\lambda, t)$*, we have that* $\mathsf{Verify}(\mathsf{pp}, x, y, \pi) = 1$ *for* $y = \mathsf{Eval}(\mathsf{pp}, x)$*.*

- ***Soundness:*** *Suppose that a* $\mathsf{poly}(t, \lambda)$*-time algorithm* $\mathcal{A}(\mathsf{pp})$ *is given the public parameters as input (for* $\mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda, t)$ *and outputs a triple* (x, y, π)*. Then, the probability that* $y \neq \mathsf{Eval}(\mathsf{pp}, x)$ and $\mathsf{Verify}(\mathsf{pp}, x, y, \pi) = 1$ *is negligible.*

- (σ, p)***-Sequentiality:*** *suppose that a* $\sigma(t)$ *parallel time algorithm* $\mathcal{A}(\mathsf{pp}, x)$ *(with* $p(\lambda, t)$*-parallelism) is given public parameters* $\mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda, t)$ *and a uniformly random input* x*. Then, the probability that* $\mathcal{A}(\mathsf{pp}, x) = \mathsf{Eval}(\mathsf{pp}, x)$ *is negligible.*

- ***Efficiency:*** *The algorithms* Setup *and* Verify *runs in time* $\mathsf{poly}(\lambda, \log t)$[18]*. The algorithm* $\mathsf{Eval}(\mathsf{pp}, x)$ *runs in parallel time* t $\mathsf{poly}(\log t, \lambda)$*-wise parallelism.*

The parameter regime of interest is when $\sigma(t) = t(1 - o(1))$ is very close to t, and $p(\lambda, t)$ is relatively large. Combining our Fiat-Shamir result (Theorems 4.1 and 4.3) with the construction of Pietrzak [Pie18], we immediately obtain our VDF result (Theorem 4.6).

Theorem 4.49 (Theorem 4.6, in more detail.)**.** *For a constant* $\epsilon > 0$*, suppose that*

- LWE *is* $\left(2^{\tilde{O}(n^{1/2})}, 2^{-n^{1-\epsilon}}\right)$*-hard, and*

- *The repeated squaring problem [RSW96] over groups of size* $2^{O(\lambda)}$ *requires* $(\sigma(t), p(\lambda, t))$ *sequential time for* $t \gg 2^{\tilde{O}(\lambda^\epsilon)}$*.*

Then, the repeated squaring function $f_{N,g}$ *can be made into a VDF with* $(\sigma(t), p(\lambda, t))$*-sequentiality. The algorithms* $(\mathsf{Setup}, \mathsf{Verify})$ *of this scheme run in time* $2^{\tilde{O}(\lambda^\epsilon)}$ *on groups of size* $2^{O(\lambda)}$ *(with* $\lambda = O(n \log n)$*). Similarly, if for some* $c > 0$*,*

- LWE *is* $\left(2^{\tilde{O}(n^{1/2})}, 2^{-\frac{n}{\log(n)^c}}\right)$*-hard, and*

- *The repeated squaring problem requires* $(\sigma(t), p(\lambda, t))$ *sequential time for* $t \gg 2^{\tilde{O}(\log(\lambda)^{c+1})}$*,*

[18] We can achieve this efficiency via complexity leveraging, but more generally allow for sub-exponential time setup and verification.

Then, $f_{N,g}$ can be made into a VDF with $(\sigma(t), p(\lambda, t))$-sequentiality. The algorithms $(\mathsf{Setup}, \mathsf{Verify})$ *of this scheme run in time $2^{\tilde{O}(\log(\lambda)^{c+1})}$. Finally, if for some $\delta > 0$,*

- LWE *is $\left(\mathsf{poly}(n), q^{-\delta n}\right)$-hard for non-uniform distinguishers, and*
- *The repeated squaring problem requires $(\sigma(t), p(\lambda, t))$ sequential time for all $t = \mathsf{poly}(n)$.*

Then, $f_{N,g}$ can be made into a VDF with $(\sigma(t), p(\lambda, t))$-sequentiality. The algorithms $(\mathsf{Setup}, \mathsf{Verify})$ *of this scheme run in time $\lambda^{O(1/\delta)}$.*

Chapter 5

Fiat-Shamir via List-Recoverable Codes (or: Parallel Repetition of GMW is Not Zero Knowledge)

5.1 Introduction

Zero-knowledge proofs, introduced by Goldwasser, Micali and Rackoff [GMR85], are a beautifully paradoxical construct. Such proofs allow a prover to convince a verifier that an assertion is true without revealing anything beyond that to the verifier. Following the introduction of zero-knowledge proofs, Goldreich, Micali and Wigderson [GMW86] constructed a zero-knowledge proof system (henceforth referred to as the GMW protocol) for the 3-coloring problem. This result is a cornerstone in the development of zero-knowledge proofs, since 3-coloring is **NP**-complete, and so the GMW protocol actually yields zero-knowledge proofs for *any* problem in **NP**.

Roughly speaking, the idea underlying the GMW protocol is for the prover to commit (via a cryptographic commitment scheme) to a random 3-coloring of the graph. The verifier chooses a random edge and the prover decommits to the colors of the two endpoints. Intuitively, the protocol is zero-knowledge since the verifier (even if acting maliciously) knows what to expect: two random different colors. An important point however is that this base protocol has poor soundness. For example, suppose that the input graph $G = (V, E)$ is not 3-colorable, but has a coloring that miscolors only one edge. In such a case, the verifier's probability of detecting the monochromatic edge is only $1/|E|$.

Thankfully, the soundness of the GMW protocol (or any other interactive proof) can be amplified by repetition. That is, in order to reduce the soundness error, one can repeat the base GMW protocol multiple times, either sequentially or in parallel, using independent coin tosses in each repetition (for both parties). At the end of the interaction the verifier accepts if and only if the base verifier accepted in all of the repetitions.

Repetition indeed reduces the soundness error, but does it preserve zero-knowledge? While it is relatively straightforward to argue that *sequential* repetition indeed preserves zero-knowledge (given the definition of *auxiliary input* zero knowledge [GO94]), this yields a protocol with a prohibitively large number of rounds. Thus, a major question in the field is whether *parallel* repetition also preserves zero-knowledge.[1]

Curiously, it has long been known that parallel repetition does not preserve zero-knowledge for some (contrived) protocols [GK96]. However, for "naturally occurring" protocols, the question remained open for decades. A sequence of recent works [KRR17,CCRR18,HL18,CCH$^+$18] showed that zero-knowledge is not preserved by repetition in very high generality (in fact, general 3-message zero-knowledge proofs can be ruled out [FGJ18]), but these works relied on extremely strong, non-falsifiable, and/or poorly understood cryptographic assumptions. The first progress on this question *based on standard assumptions* was due to Canetti *et al.* [CCH$^+$19] and Peikert and Shiehian [PS19], who showed that some *classical* ZK protocols [GMR85, Blu86] fail to remain ZK under parallel repetition. However, their results conspicuously fail to capture the GMW protocol (and indeed fail to capture "most" protocols). Thus, an answer to the following basic question has remained elusive for over 30 years [DNRS99, BLV03]:

[1] In particular, a positive resolution of this question would yield 3-message zero-knowledge proofs for all of **NP** (assuming also non-interactive commitments), thereby settling the long-standing open problem of the round complexity of zero-knowledge proofs.

Does parallel repetition of the GMW *protocol preserve zero-knowledge (under standard cryptographic assumptions)?*

As one of our main results, we answer this question in the negative, assuming the hardness of learning with errors (LWE) [Reg05].

Theorem 5.1 (Informally Stated, see Theorem 5.54)**.** *Assume that* LWE *holds. Then, there exists a commitment scheme C (in the common random string model) and a polynomial t such that t-fold parallel repetition of the* GMW *protocol (using C as its commitment scheme) is not zero-knowledge.*

We briefly make two remarks on Theorem 5.1:

- The commitment scheme C used in order to prove Theorem 5.1 is a natural one.[2] The common random string consists of a public-key of an encryption scheme (which if using a suitable encryption scheme can simply be a uniformly random string). One commits by simply encrypting messages and decommits by revealing the randomness used in the encryption.

  Still, we point out that Theorem 5.1 leaves open the possibility that parallel repetition of GMW is zero-knowledge when instantiated with a specially tailored commitment scheme.

- The number of repetitions t for which we can show that the t-fold parallel repetition of GMW has $\mathsf{negl}(n)$ soundness error, but is not zero knowledge, is $|E(G)| \cdot n^{\epsilon}$ for any $\epsilon > 0$, where $|E(G)|$ denotes the number of edges in the graph. Under the subexponential LWE assumption, the n^{ϵ} factor can be reduced to $\log^{c} n$ for some $c > 1$. This still leaves open a (very) small window of possible values for t so that the t-fold repetition of GMW is both sound and zero-knowledge (see Remark 5.55 for further discussion).

We prove Theorem 5.1 through a more general result showing that parallel repetition does not preserve zero-knowledge for a large class of protocols. This class

[2] In fact, this instantiation dates back to the original [GMW86] paper.

includes all general-purpose public-coin[3] zero-knowledge proofs for **NP** that we are aware of (when instantiated with a specific commitment scheme). In particular, this includes protocols based on the influential MPC-in-the-head paradigm [IKOS07] and more generally based on zero-knowledge PCPs (see, e.g., a recent survey [Ish20]).

All of the above negative results are shown by making *positive* progress on the closely related question of soundly instantiating the prolific Fiat-Shamir heuristic, which is our main focus, and is discussed next.

5.1.1 Securely Instantiating Fiat-Shamir

The Fiat-Shamir heuristic [FS87] is a generic technique for eliminating interaction in *public-coin* interactive proofs.[4] This technique has been extremely influential both in practice and in theory.

Consider for example a 3-message public-coin interactive proof that $x \in L$. In such a protocol first the prover sends a message α, the verifier responds with random coins β and finally the prover sends the last message γ. The basic idea underlying the Fiat-Shamir heuristic is to replace the random coin tosses β of the verifier by applying a hash function to the the transcript thus far, i.e., by setting $\beta = h(x, \alpha)$. Since the prover can now compute the verifier's coin tosses, the entire interaction consists of having the prover send the message (α, β, γ) in one shot.

It has been long known that the Fiat-Shamir heuristic is sound when the hash function is modeled as a *random oracle* [BR93, PS96, BCS16]. In reality however, we need to realize the hash function with a concrete cryptographic hash function. Following [CCH$^+$19], we say that a hash function family $\mathcal{H}$ is FS-$\mathsf{compatible}$[5] with a (public-coin) interactive protocol Π, if applying the Fiat-Shamir transform to Π, with a random choice of $h \in \mathcal{H}$, yields a computationally sound argument system. A

[3] Recall that an interactive proof is *public-coin* if all the verifier does throughout the interaction is simply toss random coins and immediately reveal them to the prover.

[4] The original goal in [FS87] was to efficiently compile (interactive) identification schemes into signature schemes, but the technique is applicable to more general protocols.

[5] We remark that the term "FS-compatible" has a different meaning in a recent work of [JKKZ21]. More specifically, [JKKZ21] defines "FS-compatibilty" to be a property of a *protocol* Π; their property consists of technical conditions that suffice for their specific hash family to instantiate FS for Π.

central problem in cryptography is to construct FS-compatible hash functions for a variety of interactive protocols of interest, thereby making them non-interactive.

While designing FS-compatible hash function families is an extremely important goal in its own right, Dwork, Naor, Reingold, and Stockmeyer [DNRS99] also showed that the existence of an FS-compatible hash function family for a (public-coin) interactive proof Π for a language $L \notin \mathbf{BPP}$, is *equivalent* to Π *not* being zero-knowledge.[6] This means, in particular, that in order to prove Theorem 5.1, it suffices to construct an FS-compatible hash function for the GMW protocol.

For a long time almost all results on instantiating Fiat-Shamir were negative [CGH98, Bar01, GK03, BDG$^+$13]. However, a recent line of work [KRR17, CCRR18, HL18, CCH$^+$19, PS19, BKM20, LV20a, JKKZ21] has made substantial *positive* progress, culminating in secure realizations of Fiat-Shamir in certain (important) cases, based on standard cryptographic assumptions.

In particular, a combination of the results of [CCH$^+$19, PS19] implies the existence of hash functions, based on LWE, that are FS-compatible for a certain class of interactive proofs. More specifically (and restricting our attention to three message protocols), this class contains interactive proofs, in the CRS model, in which for every $x \notin L$ and first prover message α, the number of random coins β that could lead the verifier to accept is polynomially bounded, and moreover, there is an efficient algorithm that finds these "bad" β's (given x, α and possibly a trapdoor associated with the CRS).

Fortunately, a natural variant of Blum's [Blu86] zero-knowledge protocol for Hamiltonicity has the above property. This is due to the fact that Blum's protocol is obtained by applying parallel repetition to a base protocol which has only a *single* choice of bad randomness. Since $1^t = 1$, the number of bad random choices when the base protocol is repeated is still 1 (and this unique bad randomness can be efficiently found). Since Hamiltonicity is $\mathbf{NP}$-complete, the works of [CCH$^+$19, PS19] yielded

[6] Roughly speaking, [DNRS99] consider a malicious verifier that answers according to the Fiat-Shamir hash function. They show that a successful simulation of such a verifier can be used to decide the language.

non-interactive zero-knowledge[7] proof-systems for all of **NP**.

While the base GMW protocol has a polynomial number of bad random strings (after all, even the *total* number of verifier random strings is polynomial), in contrast to Blum's protocol, when the protocol is repeated, this number becomes *exponential*. This means that the approach of [CCH$^+$19,PS19] no longer applies. A similar problem occurs for the parallel repetition of any base protocol with more than a single bad random choice for the verifier, which is extremely common.

We emphasize that the interest in these additional zero-knowledge protocols is not purely theoretical. In particular, some of the most efficient zero-knowledge proof-systems, such as those based on the MPC-in-the-head paradigm, also do not have a polynomial set of bad randomnesses and consequently the techniques of [CCH$^+$19, PS19] are not applicable to them.

Fiat-Shamir for Commit-and-Open Protocols. Our second main result shows how to securely realize the Fiat-Shamir transformation when applied to a much broader class of interactive proofs than what was known before (including the GMW protocol). More specifically, this class consists of the "parallel repetition of any commit-and-open protocol". By a commit-and-open protocol, we basically refer to protocols that have the following structure:

1. P commits to a string w.
2. V samples random coins r and sends them to P. These random coins, together with the main input x, specify a subset S of indices of w.
3. P decommits to w_S and V accepts or rejects based on some predicate $V(x, r, w_S)$.

Note that the GMW protocol indeed fits into this framework: w is a (random) 3-coloring of the graph, the set S specifies a random edge and V simply checks that the edge is properly colored.

[7] In contrast to the discussion in the beginning of the introduction, in the context of applying Fiat-Shamir positively in order to construct *non-interactive zero-knowledge proofs*, it suffices that the base interactive proof be *honest-verifier* zero-knowledge. Honest-verifier is indeed known to be preserved under parallel repetition.

Theorem 5.2 (Informally Stated, see Theorem 5.53)**.** *Assume that* LWE *holds. Then, there exists a commitment scheme* C *(in the* CRS *model), such that for every commit-and-open protocol* Π_C *there exists a polynomial* t *and a hash function family* $\mathcal{H}$*, such that the hash family* $\mathcal{H}$ *is* FS*-compatible with the* t*-fold parallel repetition* $(\Pi_C)^t$ *of* Π_C*.*

By the connection established by [DNRS99], Theorem 5.1 follows immediately from Theorem 5.2.

Remark 5.3. *An important example of a commit-and-open protocol is Kilian's [Kil92] celebrated succinct argument-system, as well as its generalizations based on interactive oracle proofs [BCS16]. However, we point out that Theorem 5.2 is not applicable to this protocol since Kilian relies on a particular* succinct *commitment scheme (based on Merkle hashing), whereas the commitment scheme* C *that we use is inherently non-succinct.*

Indeed, the question of securely applying Fiat-Shamir to Kilian's protocol (as envisioned by Micali [Mic94]), remains a fundamental open problem (see also [GW11, BBH$^+$*19]).*

Because it applies to parallel repetitions of *all* commit-and-open protocols (rather than just those with a single bad challenge), Theorem 5.2 substantially generalizes the class of protocols that have sound Fiat-Shamir instantiations in the standard model. We believe that Theorem 5.2 (and the techniques underlying its proof) are likely to lead to new feasibility results for non-interactive cryptographic protocols in the standard model.

Fiat-Shamir for Parallel Repetition of Multi-Round Protocols. We next turn to discuss our results for *multi-round* protocols. Let Π be a public-coin multi-round interactive proof system. As above, the application of Fiat-Shamir to such a protocol simply replaces the verifier's random coin tosses in each round with a hash of the entire transcript up to that point.

When considering protocols with a large number of rounds, some care must be taken. For example, if we take the *sequential* repetition of (say) the GMW protocol and try to apply Fiat-Shamir, it is not too difficult to see that the resulting non-interactive protocol is not sound *regardless of the Fiat-Shamir hash function* (e.g., even if the hash function is modeled as a random oracle). The issue is that after the compilation, the cheating prover can effectively "rewind" the verifier to a previous state (see [BCS16] for more details).

Thus, following [CCH+19], we restrict our attention to protocols satisfying a stronger soundness condition called *round-by-round soundness.* Loosely speaking, a protocol is round-by-round (RBR) sound, if soundness holds in each round individually. In more detail, RBR soundness dictates the existence of a predicate State (which need not be efficiently computable) mapping partial transcripts to the set $\{\mathsf{accept}, \mathsf{reject}\}$ such that:

1. If $x \notin L$ then the State of the empty transcript is rejecting.

2. Given a rejecting partial transcript τ and any prover message α, with all but negligible probability over the verifier's next coin tosses β, the partial transcript $(\tau|\alpha|\beta)$ is also rejecting (where '|' denotes concatenation).

3. The verifier always rejects *full* rejecting transcripts.

Note that round-by-round soundness implies standard soundness: the protocol starts off in a rejecting state and, with high probability, will remain so until the very end in which case the verifier is required to reject. Prototypical examples of protocols satisfying round-by-round soundness include the *sumcheck protocol* [LFKN90] and the related [GKR08] protocol (see [CCH+19, JKKZ21] for details).

We say that a protocol with RBR soundness has $\mathsf{efficiently\ recognizable\ bad\ randomness}$ if given a *rejecting* partial transcript $\tau|\alpha$, ending with a prover message α, the set of verifier coins β that make $(\tau|\alpha|\beta)$ turn into an *accepting* partial transcript is efficiently recognizable (potentially also given access to a trapdoor of a CRS, if such exists).

The works [CCH+19, PS19] imply LWE-based FS-compatible hash functions for interactive proofs with *negligible* RBR soundness error in which the bad randomness is not just efficiently recognizable, but moreover the set is efficiently *enumerable* (i.e., the set of bad randomness is polynomially bounded and can be explicitly generated in polynomial time). We extend their result to protocols obtained by taking parallel repetition of an r-round base protocol with RBR soundness error *close to* $1/r$, and without any constraint on the number of choices of bad randomness.

Theorem 5.4 (Informally Stated, see Theorem 5.68)**.** *Let* Π *be a* $2r+1$*-message interactive proof with round-by-round soundness error* $\frac{1-\epsilon}{r}$ *with efficiently reconizeable bad randomness. Then, there exists a polynomial* $t = t(n, \lambda, \epsilon)$*, and a hash family* $\mathcal{H}$*, such that* $\mathcal{H}$ *is* FS*-compatible with* Π^t*.*

Remark 5.5. *Theorem 5.2 actually follows from Theorem 5.4 since constant-round protocols with negligible soundness are automatically round-by-round sound, and the specific type of commitment scheme makes the bad randomnesses efficiently computable.*

However, we set apart these two results for two reasons. First, the proof of Theorem 5.2 is simpler than that of Theorem 5.4 and suffices for many protcols of interest. Second, we are unable to achieve a tight result with respect to the number of repetitions in Theorem 5.4 as we did for Theorem 5.2.

Finally, we note that Theorem 5.4 can be combined with the main insight of [JKKZ21] (which is orthogonal to our work) to *further* generalize the class of protocols Π that have sound Fiat-Shamir instantiations. Informally, the [JKKZ21] technique of *lossy* correlation intractability allows us to additionally handle protocols where bad challenges for the i-th round can only be efficiently recognized given non-uniform advice about the *previous* rounds' challenges. For example, this allows us to instantiate Fiat-Shamir for parallel repetitions of the [GKR08] protocol, even when the field size of the base protocol is *poly-logarithmic*. In contrast, [JKKZ21] can only handle variants of [GKR08] with an exponential field size.[8] For example, this precludes applications in which one needs to materialize entire truth tables of polynomials over the field.

[8] [JKKZ21] use a large field in order to have negligible soundness error but only polynomially many bad challenges.

5.1.2 Technical Overview

We now describe our techniques for proving Theorem 5.2, with a particular focus on the GMW protocol for ease of understanding. Our starting point is the work of [CCH+19], which gave the first instantiation of Fiat-Shamir in the standard model based on standard cryptographic assumptions. As in prior work [KRR17, CCRR18, HL18], their Fiat-Shamir instantiation makes use of the framework of *correlation intractability* [CGH98], which we recall here.[9]

A hash family $\mathcal{H}$ is said to be (single input) correlation-intractable for a binary relation R if it is computationally hard, given a hash key $h \leftarrow \mathcal{H}$, to find a "correlation", i.e., an input x such that $\big(x, h(x)\big) \in R$. Such a security property is plausibly instantiable, and is satisfied by a random oracle, whenever the relation R is *sparse*, meaning that for any input x, the fraction of outputs y for which $(x, y) \in R$ is negligible.

Despite this plausibility argument, and despite the intriguing connection to Fiat-Shamir in the standard model (which we will see in a moment), there were essentially no instantiations of correlation intractability (beyond very simple relations such as those for which $(x, y) \in R$ if and only if $y = c$ for a constant c) before 2016. However, a flurry of recent works (including [CCR16, KRR17, CCRR18, HL18, CCH+19, PS19, LVW19, BFJ+20, GJJM20, LNPT19, BKM20, LV20a, JKKZ21, LNPY20, LV20b]) have (1) instantiated various flavors of correlation-intractable hash functions based on plausible cryptographic assumptions and (2) applied these hash functions to achieve independently useful cryptographic goals.

We discuss this line of work in detail in Section 5.1.4, but for now, we recall the following result from [PS19], which is most relevant for our purposes. It is a construction of correlation intractability for *functions*: we say that $\mathcal{H}$ is CI for a function f if it is CI for the relation $R_f = \{(x, f(x))\}$.

Theorem 5.6 ([PS19], informal)**.** *Under the* LWE *assumption, there exists a hash family* $\mathcal{H}$ *that is correlation intractable for all functions that are computable in (a priori bounded) polynomial time.*

[9] In fact, [DNRS99] cites personal communication with Chaum and Impagliazzo for an early variant of this connection. Full formalizations of this paradigm appear in [CCRR18, CCH+19].

As described in the theorem statement, Theorem 5.6 has the following two limitations (which are also present in the predecessor work [CCH+19][10]).

- They only achieve security for relations $R \subseteq X \times Y$ that represent *functions.* That is, for every $x \in X$ there is (at most) a single $y \in Y$ such that $(x, y) \in R$.
- They require that the functions are *efficiently computable.*

Both of these drawbacks turn out to be relevant for Fiat-Shamir instantiations. To see this, we first discuss how CI relates to the instantiation of Fiat-Shamir for interactive proofs. For simplicity, we focus on the task of compiling 3-message public coin interactive proofs. Such protocols have the following syntax.

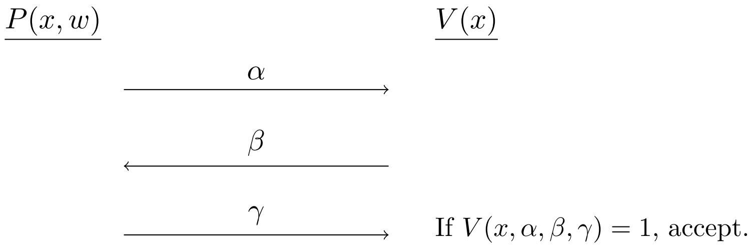


Figure 5-1: A 3-message public coin interactive proof Π.

After applying the Fiat-Shamir transform using hash family $\mathcal{H}$, we obtain the protocol $\Pi_{\mathrm{FS},\mathcal{H}}$ below.

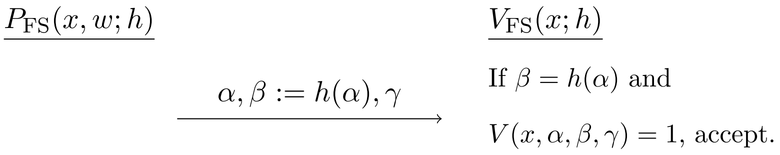

$P_{\mathrm{FS}}(x, w; h)$
$V_{\mathrm{FS}}(x; h)$
$\alpha, \beta := h(\alpha), \gamma$
If $\beta = h(\alpha)$ and
$V(x, \alpha, \beta, \gamma) = 1$, accept.

Figure 5-2: The Protocol $\Pi_{\mathrm{FS},\mathcal{H}}$.

In this situation, consider the following relation $R^{(0)} = R^{(0)}_{x,\Pi}$ for a false statement x, which we call the (naive) bad-challenge relation for Π:

$$R^{(0)}_{x,\Pi} = \{(\alpha, \beta) : \exists\gamma \text{ s.t. } V(x, \alpha, \beta, \gamma) = 1\}.$$

[10] More specifically, this limitation is present in the subset of results in [CCH+19] that are based on quantitatively standard cryptographic assumptions

It follows almost syntactically that if $\mathcal{H}$ is CI for $R^{(0)}_{x,\Pi}$ (for all false statements x), then $\mathcal{H}$ soundly instantiates Fiat-Shamir for Π. Thus, the problem of instantiating Fiat-Shamir is reduced to constructing sufficiently general-purpose correlation intractable hash functions. Bearing in mind the two drawbacks of Theorem 5.6, it is worth noting that $R_{x,\Pi}$ is (in general) not even a function, let alone an efficiently computable one.

Fiat-Shamir for GMW**.** With the above background in mind, we turn to the task at hand: finding a Fiat-Shamir instantiation for the parallel repeated GMW protocol. Abstractly, a t-wise parallel repetition of a protocol Π has the following syntax.

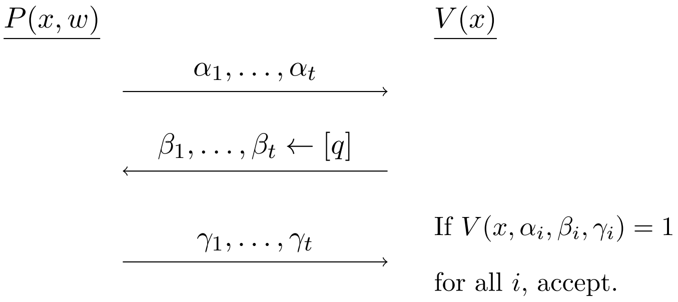

$\underline{P(x,w)}$
$\underline{V(x)}$
$\alpha_1, \ldots, \alpha_t$
$\beta_1, \ldots, \beta_t \leftarrow [q]$
$\gamma_1, \ldots, \gamma_t$
If $V(x, \alpha_i, \beta_i, \gamma_i) = 1$
for all i, accept.

Figure 5-3: A parallel-repeated protocol Π^t.

In the case of GMW, the input x is a graph $G = (V, E)$, the witness w is a 3-coloring of G, the messages α_i are commitments to (a random shuffling of the colors of) w, each $\beta_i = (u_i, v_i) \in E(G)$ specifies a randomly selected edge, and the γ_i are decommitments[11] (z_i, r_i) to the colors $z_i = (w(u_i), w(v_i))$. The verification procedure checks that the decommitments are all valid and that each (revealed) colored edge is not monochromatic. Note that the "alphabet size" q denotes the size of the the verifier's challenge space, which in this case is $q = |E|$.[12]

Recall that by Theorem 5.6, we would be done if (1) the relation $R^{(0)} = R^{(0)}_{x,\Pi^t}$ above represented a function f, and (2) the function f were efficiently computable.

[11] A decommitment (m, r) of a string com is a message m and choice of commitment randomness r such that $\mathsf{com} = \mathsf{Com}(m; r)$.

[12] Our results in this overview may appear to require that q is polynomial in n, but we show in Section 5.3.2 how to reduce from general q to polynomial-size q via *subsampling*. This allows us to handle Fiat-Shamir for parallel repetitions of arbitrary commit-and-open protocols.

As a first step, we show (following [HL18, CCH+19]) how to replace the relation $R^{(0)}$ with a relation R that is *efficiently verifiable*, i.e., there is an efficient algorithm that *recognizes* bad challenges.

In a nutshell, the "commit-and-open" structure of the GMW protocol allows us to replace the "naive bad-challenge relation" $R^{(0)}_{x,\Pi^t}$ with the relation

$$R_{x,\Pi^t} := \Big\{ \big((\alpha_1, \dots, \alpha_t), (\beta_1, \dots, \beta_t) \big) : \text{ each } z_i := \mathsf{Extract}(\alpha_i[\beta_i]) \text{ has two distinct colors} \Big\},$$

where $\mathsf{Extract}$ denotes a function that extracts a committed bit b from a commitment com. In other words, the relation $R_{x,\Pi}(\alpha, \beta)$ can be verified by extracting from $\alpha[\beta]$ the appropriate committed string z and then checking whether the two colors defined by z are distinct. If the commitment scheme is efficiently extractable (given a trapdoor; e.g., this holds if Com is the encryption algorithm of a public-key encryption scheme), then R_{x,Π^t} can be efficiently verified. Thus, to instantiate Fiat-Shamir for this (natural) instantiation of the GMW protocol, it suffices to construct a hash family $\mathcal{H}$ that is CI for this particular (efficiently verifiable) relation R_{x,Π^t}.

The Problem: Too Many Bad Challenges. The main barrier to instantiating Fiat-Shamir for GMW is due to the *first* drawback of the [CCH+19, PS19] results, namely, that R is *not* a function. We quantify the extent to which R is not a function with the following terminology.

Definition 5.7 (d-Bounded Relation)**.** *We say that a relation $R \subseteq \{0,1\}^n \times \{0,1\}^m$ is $d = d(n)$-*bounded *if $|R(x)| \leq d$, for all $x \in \{0,1\}^n$, where $R(x) = \{y \in \{0,1\}^m : (x,y) \in R\}$.*

We focus on *absolute* rather than *relative* boundedness (aka density) due to the limitations of prior work on instantiating correlation intractability. In particular, the CI hash families of [CCH+19, PS19] were shown to satisfy correlation intractability for (efficiently computable) *functions*, i.e., 1-bounded relations. In prior work [CCH+19, JKKZ21], CI for relations that are *not* functions was only achieved in a very limited sense: for d-bounded relations R, it is noted that a hash family $\mathcal{H}$ that is CI for efficiently computable functions *with* $\frac{1}{d}$ *quantitative security* is also CI for d-bounded relations that are "efficiently enumerable".[13] This is proved via a trivial "guessing" reduction from CI for functions with a security loss of $\frac{1}{d}$. In prior works, only polynomial (or slightly superpolynomial) values of d were considered for this reason.

However, in the case of parallel repeated GMW, the relation $R = R_{x,\Pi^t}$ may be only $(|E(G)| - 1)^t$-bounded. In other words, for every $\alpha = (\alpha_1, \ldots, \alpha_t)$, there may be $(|E(G)| - 1)^t$ challenges β such that $(\alpha, \beta) \in R_{x,\Pi^t}$. As a result, the "guessing reduction" above incurs a security loss that is exponential in the security parameter, resulting in a useless reduction. Achieving CI for d-bounded relations for *large* values of d – and instantiating Fiat-Shamir for protocols with *many* bad challenges – was an unsolved problem.

Main Idea: Derandomization. Our high-level idea for resolving this problem is using *derandomization* to reduce the *effective* d-boundedness of the relation R. Namely, we employ a two-step process.

1. Devise a randomness-efficient procedure for sampling challenges $(\beta_1, \ldots, \beta_t) \leftarrow \mathsf{Samp}(r)$ such that only *polynomially* many bad choices of r lead to bad challenges (for any given pair (x, α)). Note that we need to do so while maintaining *negligible* soundness error. That is, we want the set of bad challenges to have *absolute* size that is polynomial, while its *relative* size (or density) is negligible.

2. Compose the sampling procedure with a hash family $\mathcal{H}_{\text{inner}}$ that is CI for polynomially-bounded relations. In particular, $\mathcal{H}_{\text{inner}}$ must satisfy CI for a new relation $\tilde{R} := \tilde{R}_{x,\Pi,\mathsf{Samp}}$ that depends on the procedure Samp as well as Π.

[13] A d-bounded relation R is *efficiently enumerable* if there is an efficient algorithm that, on input x, explicitly generates the set of all y such that $(x, y) \in R$.

This process yields a correlation-intractable hash family for R by a natural composition. Namely, our hash family will consist of hash functions h' defined as

$$h'(x) = \mathsf{Samp}(h(x))$$

where $h \leftarrow \mathcal{H}_{\text{inner}}$ comes from a previously constructed CI hash family (namely, the families from [CCH+19, PS19]).

Another interpretation of our approach is that we instantiate Fiat-Shamir for a (parallel repeated) protocol Π^t by implicitly working with a *derandomized parallel repetition*[14] of Π.

Still, several crucial details remain unclear from this outline:

- How should we instantiate the sampling procedure Samp?
- How do we prove that the resulting hash family $\mathcal{H}'$ is FS-compatible for Π^t?

Indeed, standard derandomization techniques such as expander walks and pseudorandom generators turn out *not* to suffice for our application, as we elaborate below. Instead, we need a *new derandomization technique*: our main technical contribution is a special-purpose instantiation of Samp and proof of security for $\mathcal{H}'$.

Naive Idea: Use a PRG. As a first (flawed) attempt to solve our problem, one might consider setting $\mathsf{Samp}(r) = G(r)$ for some pseudorandom generator G (either cryptographic [BM82] or "Nisan-Wigderson style" [NW88, IW97, AK97]; indeed, the PRG would only have to fool a specific test related to Π). We briefly describe why this approach fails:

- **The new relation $\tilde{R}$ is *still* not bounded enough**. To understand this point, we need to specify what tests the PRG G has to fool. By staring at the problem, we see that G should have the property that for every statement x

[14] The type of derandomization that we require is related to, but different from, the "sampler-based" [Gol11, Vad12] derandomized parallel repetition of Bellare, Goldreich and Goldwasser [BGG90]. The exact approach of [BGG90] does not work for us for reasons similar to the "naive" PRG approach below.

and first messages $\alpha_1, \dots, \alpha_t$, the probability that $G(r) = (\beta_1, \dots, \beta_t)$ has the property that $(\alpha, G(r)) \in R_{x,\Pi}$ is close to the sparsity of R. Unfortunately, known PRG constructions still have the property that the *absolute* number of such "bad r" is exponential in the seed length,[15] while we need this number to be polynomial in the seed length.

- **The new relation $\tilde{R}$ is not efficiently enumerable**. On top of parameter issues, the relation $\tilde{R}$ constructed in step (2) above seems hard to compute, because it syntactically requires computing preimages (of exponential-size sets!) under the map G. Indeed, the relation $\tilde{R}$ has the form:

$$\tilde{R}_{x,\Pi,G} = \{(\alpha, r) : (\alpha, G(r)) \in R_{x,\Pi^t}\},$$

so the set of all r such that $(\alpha, r) \in \tilde{R}_x$ is $G^{-1}(\{\beta : (\alpha, \beta) \in R_x\})$. Since $\tilde{R}$ does not seem to be efficiently enumerable, we do not know how to construct a CI hash family for it.

Our Code-Based Derandomization. Since the naive idea of using a PRG for derandomization fails, we now study our special-purpose derandomization problem in more detail. In particular, we crucially take advantage of the *parallel repetition structure* of the relation R_{x,Π^t} to reframe the problem.

As above, our plan is to use some function $\mathsf{Samp}(r) \to (\beta_1, \dots, \beta_t)$ along with a hash family $\mathcal{H}$ that is correlation intractable for the relation $\tilde{R}$, which can be expressed as

$$\tilde{R}_{x,\Pi^t,\mathsf{Samp}} = \Big\{(\alpha, r) : (\alpha_i, \mathsf{Samp}(r)_i) \in R_{x,\Pi} \text{ for all } i\Big\}.$$

Moreover, for each fixed pair (x, α_i), we know that the collection S_i of all β_i such that $(\alpha_i, \beta_i) \in R_{x,\Pi}$ is *not too large*: if the protocol Π has soundness error $1 - \epsilon$ (meaning

[15] This boils down to the suboptimal ϵ-dependence of the seed length of known PRGs that are ϵ-pseudorandom. In order for the number of "bad r" to be polynomial, we would need a PRG with seed length $O(\log m) + \log(1/\epsilon)$ – that is, we cannot afford any constant $c > 1$ in front of the $\log(1/\epsilon)$ term).

that cheating provers are caught with probability ϵ; in the case of GMW, we have $\epsilon = \frac{1}{|E(G)|}$), then $|S_i| \leq (1-\epsilon)q$ for all i (recall that q denotes the verifier's challenge space in the base protocol).

More abstractly, we are interested in relations of the form

$$\tilde{R}_{x,\Pi^t,\mathsf{Samp}} = \Big\{(\alpha, r) : \mathsf{Samp}(r)_i \in S_i \text{ for all } i\Big\},$$

where:

- Each set $S_i \subseteq [q]$ is promised to have some bounded size $|S_i| \leq (1-\epsilon)q$,
- Each set S_i can be efficiently computed from (x, α). (This property is guaranteed by the efficient verifiability of R).

Since our hope is to use $\mathcal{H}$ from [CCH+19, PS19] – which is only CI for *efficiently enumerable* relations – we have two strong demands of the procedure $(\beta_1, \ldots, \beta_t) \leftarrow \mathsf{Samp}(r)$:

- For all x and all α, the number of r such that $\mathsf{Samp}(r) \in S_1 \times \ldots \times S_t$ should be *polynomial* in the length of r.
- Moreover, the (polynomial-size) set of all such r should be be efficiently computable given (x, α) (or, essentially equivalently, the sets $S_1, \ldots S_t$).

Almost miraculously, if we think of our sampler Samp as the encoding procedure Encode of an error-correcting code, this set of requirements *exactly corresponds* to an important notion in coding theory: (errorless) list recovery [GI01]!

We now (informally) recall the definition of an (error-free) list-recoverable code. Let $\mathsf{Encode} : \{0,1\}^\lambda \to [q]^t$ denote an efficient encoding procedure. We say that $(\mathsf{Encode}, \mathsf{Recover})$ is a (ℓ, L)-list recoverable code if

- For all sets (called input lists) $S_1, \ldots, S_t$ of size at most ℓ, the number of messages $m \in \{0,1\}^\lambda$ such that $\mathsf{Encode}(m) \in S_1 \times \ldots \times S_t$ is at most L, and

- The algorithm $\mathsf{Recover}(S_1, \ldots, S_t)$, given descriptions of the input lists $S_1, \ldots, S_t$, efficiently returns the $\leq L$ corresponding messages (called the $\mathsf{output\ list}$).

List-recoverable codes were introduced by [GI01] as a tool for constructing more efficient list-decodable codes. For our application, we define $\mathsf{Samp}(r) := \mathsf{Encode}(r) \in [q]^t$, so that

- The *alphabet* q of the code is exactly the challenge space for the base protocol Π.

- The *block-length* t of the code is the *number of repetitions* of the protocol Π,

- The *input list* size $\ell = (1 - \epsilon)q$ corresponds to the *boundedness* of the relation R_Π, and

- The *output list* size L is a bound on the number of seeds r that are mapped to bad challenges, and so should be some polynomial in the security parameter λ.[16]

We emphasize that the parameter regime we are interested in is *qualitatively different* than is typical in coding theory. In the coding theory literature (see [HW15a, Figure 1] as well as [RW18] for examples), the input list size ℓ is typically very small[17] compared to the alphabet size q, while the parameters they want to optimize are the block-length t (ideally $t = O(\lambda)$), as well as the output list size L (which is important for efficient decoding when the list-recoverable code is used as a component in a larger construction).

On the other hand, our setting has a very large value of ℓ (potentially as high as $(1 - \epsilon)q$); we then want to optimize for the block-length t, which is ideally not much larger than $1/\epsilon$, but multiplicative factors of $\mathsf{poly}(\lambda)$ do not really bother us (in particular, the code can have rate $o(1)$). Meanwhile, the output list size L is not too important for us (as long as it is polynomial), but it is crucial that list-recovery

[16] The dependence is actually allowed to be $\mathsf{poly}(\lambda, q, 1/\epsilon)$

[17] For example, degree k Reed-Solomon codes over $\mathbb{F}_q$ can handle $\ell \leq \frac{q}{k}$, while known higher rate constructions can only tolerate much smaller values of ℓ.

is computationally efficient (rather than information-theoretic), which differs from many prior works.

As described above, there is a tight connection between list-recoverable codes and correlation-intractable hash families through the construction $h'(x) = \mathsf{Encode}(h(x))$:

Theorem 5.8 (Informally stated, see Theorem 5.36)**.** *Suppose that*

- *$\mathcal{H}$ is a hash family that is CI for efficient functions,*
- *$R = R_{x,\Pi}$ is an* efficiently verifiable *relation with output space $[q]$ and sparsity $1 - \epsilon$, and*
- *$(\mathsf{Encode}, \mathsf{Recover})$ is a $((1-\epsilon)q, L)$-list recoverable code mapping $\{0,1\}^\lambda \to [q]^t$.*

Then, the hash family defined by $h'(x) = \mathsf{Encode}(h(x))$ is CI for the relation R_{x,Π^t}, and is therefore FS*-compatible with the protocol Π^t.*

In Section 5.3.1, we rephrase Theorem 5.8 fully in the language of correlation intractability (without reference to any protocol Π) by defining a natural notion of "product relation". We then show that list-recoverable codes can be used to generically construct CI for product relations from CI for functions. Then, in Sections 5.5 and 5.6, we show how this form of CI allows us to prove our general FS results: Theorem 5.2 and Theorem 5.4. For the generalization to many-round protocols, we in fact make use of *error-tolerant* (rather than error-free) list-recoverable codes.

Final Step: Constructing the Codes. However, an important question remains: do there actually exist codes satisfying all of the properties that we need? To summarize (for the case of 3-message protocols), we want the following conditions to hold for a code defined by $\mathsf{Encode} : \{0,1\}^\lambda \to [q]^t$.

1. The code should be (ℓ, L)-list recoverable for $\ell = (1-\epsilon)q$ and $L = \mathsf{poly}(q/\epsilon)$.
2. Both encoding and list recovery should be *computationally efficient* rather than information-theoretic.

3. Subject to (1) and (2), the block-length t should be as small as possible.

Conditions (1) and (2) are necessary to obtain any valid Fiat-Shamir instantiation for some sufficiently large number of (parallel) repetitions of a protocol Π, while condition (3) seeks to minimize the number of repetitions (hopefully to a number not much larger than what is required in the interactive setting).

It is not difficult to argue that a random code $f : \{0,1\}^\lambda \to [q]^t$ satisfies condition (1) with high probability, with t indeed on the order of $1/\epsilon$ (see Theorem 5.40); however, it (of course) does not satisfy condition (2). On the other hand, known list-recoverable codes with *efficient* list-recovery are only designed to handle small input list sizes. This includes algebraic codes [GS98, PV05, GR08], expander codes [SS94, HW15a], and codes built by a combination of these tools [GI01, GI02, GI03, GI04]. As mentioned before, prior work did not primarily optimize for the *input list sizes*. In fact, aside from some of the works on algebraic codes, the parameter settings in prior work require $\ell = q^{o(1)}$;[18] these prior works were instead mostly focused on achieving high rate and very efficient algorithmic encoding/recovery.

In this work, we give a randomized construction of a code satisfying our demands via *code concatenation* [For66] combining an *algebraic code* with a *random code* (in a parameter regime where brute force decoding is polynomial-time). This is similar to the approach of [GI01] (although they use random "pseudolinear" codes rather than truly random codes for reasons of efficiency), but the parameters of our code concatenation (i.e. the relationship between the algebraic code's parameters and the random code's parameters) are quite different from [GI01].

Code concatenation is a technique based on the following simple idea: given two codes $C_{\text{out}}, C_{\text{in}}$ such that *alphabet symbols* of C_{out} can be interpreted as messages for C_{in}, it is possible to encode a message m by first computing $y = \mathsf{Encode}_{\text{out}}(m)$ and then encoding each symbol y_i using C_{in}. Code concatenation admits simple composition theorems for list-recovery, so the main question is whether there are parameter settings for $C_{\text{out}}, C_{\text{in}}$ that meet our demands.

[18] An interesting concurrent and independent work [DW20] uses expander code-based techniques to construct a variant of list-recoverable codes with constant rate and $\ell = q^{\Omega(1)}$, but this is still far from the parameter regime that we care about.

It turns out that by setting the alphabet size q' of the outer code to be polynomially larger than the alphabet size of the inner code (which is q), the concatenation $C_{\text{out}} \circ C_{\text{in}}$ can be shown to be list-recoverable for large input list sizes as long as the outer code is list-recoverable for *moderately large* input list sizes. Moreover, list-recovery is efficient even if the *inner* code must be list-recovered by brute force; this allows for the input list size for $C_{\text{out}} \circ C_{\text{in}}$ to be very large (as this parameter is inherited from C_{in}). In the end, our choice of C_{out} is a Parvaresh-Vardy code with carefully chosen parameters to optimize for the block-length t of the final construction:

Theorem 5.9 (Informal, see Lemma 5.45)**.** *For all $\ell < q = \mathsf{poly}(\lambda)$, there exists a probabilistically constructable family of codes*

$$\left\{ C : \{0,1\}^{\lambda} \to [q]^{\lambda^2 \cdot \frac{\log(\lambda)}{\log(q/\ell)}} \right\}$$

that is $\Big(\ell, \mathsf{poly}(\lambda)\Big)$-list recoverable with all but $2^{-\lambda}$ probability.

In particular, for $\ell = (1-\epsilon)q$, we obtain block-length $t = \tilde{O}(\lambda^2/\epsilon)$. We refer the reader to Sections 5.4, 5.5.1 and 5.6.1 for more details.

5.1.3 Reflections: Fiat-Shamir via Coding Theory

In summary, our main technique relates correlation intractability for *relations* to correlation intractability for *functions* in two high-level steps.

1. **List Recoverable Codes**. Given a protocol Π whose bad challenges are (approximate) product sets $S = S_1 \times \dots S_t \subseteq [q]^t$ (such as those arising from parallel repetition), we construct a code $C : \{0,1\}^{\lambda} \to [q]^t$ that *avoids* all such S: namely, every product set S contains only polynomially many codewords $C(m)$.

2. **Composition**. We prove that such codes *compose* with a hash family $\mathcal{H}$ that is CI for functions to obtain a hash family $C \circ \mathcal{H}$ that is CI for product relations.

One can view this as a special case of a more general paradigm: given the results of [CCH+19, PS19], we can reduce the problem of instantiating Fiat-Shamir for *any* public-coin interactive proof to a coding-theoretic problem. For example, given a constant-round (or more generally, round-by-round sound) interactive proof Π for a language $\mathcal{L}$, soundness guarantees that for every transcript prefix τ of Π on an input $x \notin \mathcal{L}$ there is a sparse set S_τ of "bad" verifier messages. We would like to construct a code $C : \{0,1\}^\lambda \to [q]$ such that C "evades" S_τ in the sense that there are at most polynomially many messages m for which $C(m) \in S_\tau$, and furthermore there is a polynomial-time algorithm that enumerates all such m. Given such a code C, the composition of the [PS19] hash function with C instantiates Fiat-Shamir for Π (assuming LWE).

For general interactive proofs, the sets S_τ may be extremely complex and decoding seems intractable. In our results above, we took advantage of the following structure of Π that makes decoding feasible:

- Π is a *parallel repetition*, which ensures that each set S_τ is a product set;
- Moreover, the base protocol has *efficiently recognizable* bad challenges.

We were then able to leverage highly non-trivial existing algorithms [GS98, PV05] to solve the resulting coding problem.

An interesting direction for future work is whether other forms of efficient decoding can be used to instantiate Fiat-Shamir for other natural protocols.

5.1.4 Related Work

Correlation Intractability and Fiat-Shamir.

We survey the recent constructions of correlation intractable (CI) hash families [CCR16, KRR17, CCRR18, HL18, CCH$^+$19, PS19, BKM20] for comparison with our work. These constructions roughly fall into two categories:

CI for Large Classes of Relations based on Non-Standard Assumptions. The initial works [CCR16, KRR17, CCRR18, HL18, CCH$^+$18] constructed hash families that achieve correlation intractability for very broad classes of relations, but they can only prove security based on strong and non-standard cryptographic assumptions. In more detail,

- [CCR16] constructs a hash family that is CI for all *efficiently verifiable* relations (i.e., relations R such that it is efficiently decidable whether $(x, y) \in R$) assuming (sub-exponentially secure) indistinguishability obfuscation (iO) as well as input-hiding obfuscation for evasive circuits [BBC+14].

- [KRR17, CCRR18] construct hash families that are CI for *all* (even hard-to-decide) sparse relations. To do so, they make assumptions that are both extremely quantitatively strong and non-falsifiable [Nao03, GW11]. For example, [CCRR18] assumes the existence of an encryption scheme such that key-recovery attacks, given (even inefficiently generated) key-dependent-message (KDM) ciphertexts, cannot succeed with probability significantly better than random guessing. [KRR17] makes a simiar assumption, and additionally assumes (subexponentially secure) iO.

- [HL18] constructs a hash family that is CI for all "efficiently sampleable relations" (similar in spirit but technically incomparable to "efficiently verifiable relations" as in [CCR16]) assuming (subexponentially secure) iO and optimally secure one-way functions—that is, a one-way function f with no inversion attacks that are significantly better than random guessing. [CCH+19] (see [CCH+18]) also gives constructions of such a hash family under "optimally secure" variants of the learning with errors (LWE) assumption (without iO).

To summarize, these hash families achieve strong notions of CI (which suffice to instantiate Fiat-Shamir for broad classes of interactive proofs) at the cost of highly non-standard assumptions.

CI for Efficient Functions based on Standard Assumptions Beginning with the work of [CCH+19] (see [CLW18]), a sequence of works [CCH+19, PS19, BKM20] gave constructions of restricted forms of correlation intractability based on widely accepted assumptions. In more detail,

- [CCH+19,PS19] construct hash families that are CI for all *efficiently computable functions*, that is, for relations R such that $(x, y) \in R \iff y = f(x)$ for some efficiently computable function f. [CCH+19] constructs such a hash family under circular-secure fully homomorphic encryption, while [PS19] relies on the plain LWE assumption.

- [BKM20] constructs hash families that are CI for *low-degree polynomial functions* based on any one of various assumptions including LWE, the decisional Diffie-Hellman (DDH) assumption, and the Quadratic Residuosity (QR) assumption. In fact, their hash families are CI for *relations* R that are "efficiently approximable" by low-degree polynomials over $\mathbb{F}_2$, i.e., relations R such that $(x, y) \in R \iff y$ is close to $p(x)$ in Hamming distance.

To summarize, these works construct hash families that are CI for (classes of) *efficient functions* (rather than relations), possibly up to some error tolerance on bits of the output.[19] To emphasize even further, there are two main drawbacks to these CI constructions:

1. They only achieve security for relations $R \subseteq X \times Y$ that represent *functions* (possibly tolerating some error).

2. They require that the functions (or, equivalently, the relations) are *efficiently computable.*

In the context of FS-compatibility, what this means is that prior work has successfully constructed hash families that are FS-compatible with interactive proofs Π whose bad-challenge relations $R_{x,\Pi}$ can be interpreted as *efficient functions.*[20] The 3-message protocols whose bad-challenge relations are (possibly inefficient) functions are those satisfying "special soundness": for every false statement x and every prover message α, there is *at most one* choice of challenge β such that an accepting proof of

[19] Indeed, the constructions of [CCH+19, PS19] also support a kind of error tolerance, although this was irrelevant for their purposes.

[20] For 3-message protocols, these are abstracted as "trapdoor Σ-protocols" in [CCH+19].

the form (α, β, γ) exists. Proof systems satisfying this notion include important protocols such as [GMR85, Blu86, FLS90], but a "typical" protocol Π will be extremely far from satisfying this notion. By a "random guessing" reduction, is it not hard to handle protocols Π that have only *polynomially many* bad challenges β for any fixed α, but again, this captures only a small class of protocols.

Finally, we note that while drawback (2) has been circumvented to a small extent in later works [LV20a, JKKZ21], some form of efficiency requirement has been necessary for all bad-challenge functions of protocols Π with Fiat-Shamir instantiations under standard assumptions. As in prior work [CCH$^+$19, PS19, BKM20], we instead work with protocols Π such that (a relaxation of) the relation $R_{x,\Pi}$ can be efficiently verified *given a trapdoor* td. In the case of [GMW86], this is achieved by using a commitment scheme with a *trapdoor* that can extract committed bits (i.e., a public-key encryption scheme).

One might wonder whether it is possible to directly show that the CI hash families of [CCH$^+$19, PS19] are also CI for relations such as $R_{x,\Pi_{\mathrm{GMW}}}$. The intuitive reason this appears to be hard is as follows: to show that the [CCH$^+$19, PS19] hash families $\mathcal{H}$ are CI for a function f, they show that a hash function $h \leftarrow \mathcal{H}$ is computationally indistinguishable from a hash function (distribution) h_f that on input x internally (1) computes $f(x)$ and then (2) outputs a value y that specifically avoids $f(x)$. It is possible to extend this proof to make $\mathcal{H}$ "avoid" a polynomial number of evaluations $f_1(x), \ldots, f_k(x)$ (by internally computing *all* of them), but for our relations of interest, the number of $(x, y) \in R$ (for a fixed x) can be close to 2^m (for $m = |y|$)! As a result, proving that the [CCH$^+$19, PS19] hash functions satisfy this form of correlation intractability appears out of reach for current techniques.

CI for Approximable Relations We note that in order to instantiate Fiat-Shamir for round-by-round sound protocols(Section 5.6), we implicitly rely on (and construct) hash families that are correlation intractable for *approximations* of a relation R in a sense similar to the abstraction introduced in [BKM20]. However, in our setting, we think of hash outputs as elements of $[q]^t$ and our metric of interest is Hamming distance in the space $[q]^t$; correspondingly, our security requirement is stronger, in that we want CI for even extremely poor approximations of R (i.e. distance significantly greater than $\frac{1}{2}$). We achieve this notion of CI when R is any (sufficiently bounded) product relation using error-tolerant list-recoverable codes.

List-Recoverable Codes and Cryptography

List-recoverable codes have previously been used [MT07,DS11,HIOS15,KNY18,BKP18] in cryptography in the context of *domain extension* [Mer88] for hash functions. That is, given a hash function $h : \{0,1\}^n \to \{0,1\}^m$, their goal is to construct another hash function $H : \{0,1\}^* \to \{0,1\}^m$ while preserving security properties such as collision-resistance. In particular we highlight the work of [HIOS15] who use list-recoverable codes to construct hash functions H that are *indifferentiable* from random functions (if h is modeled as a random oracle). In their construction (as well as in [MT07,DS11]), it suffices to use off-the-shelf Parvaresh-Vardy codes [GUV09], albeit in somewhat non-standard parameter regimes. For example, [HIOS15] considers a regime with (1) subexponential (rather than polynomial) time list-recovery and (2) input list sizes of size q^δ for some $0 < \delta < 1$ (and q is the alphabet size).

One notable difference between our use of list-recoverable codes as compared to [MT07, DS11, HIOS15, KNY18, BKP18] is that in the context of domain extension, *precomposition* with a list-recoverable code (i.e. encoding the input x and then hashing it) is the technique used; on the other hand, we *post-compose* a hash function h with a code (i.e. we encode the *output* $h(x)$) in order to facilitate a kind of “output compression” (rather than domain extension).

5.2 Preliminaries

5.2.1 Interactive Proofs and Zero-Knowledge

Definition 5.10. *An* interactive proof for a language L with completeness error $c = c(n)$ and soundness error $s = s(n)$ *consists of a probabilistic polynomial-time interactive* *verifier V such that:*

- *(Completeness:) If $x \in L$ then there is an interactive function P_x such that $V(x)$, when interacting with P_x, accepts with probability at least $1 - c(|x|)$.*

- *(Soundness:) If $x \notin L$ then $V(x)$, when interacting with* any *(even computationally unbounded) interactive function P^*, accepts with probability at most $s(|x|)$.*

If the error parameters c or s are omitted, by default we require them to be negligible functions. If $c = 0$, we say that the proof-system has perfect completeness.

One of the main metrics of a proof system is the number of messages m exchanged between the prover and the verifier before the verifier decides whether or not to accept. Typically, this depends only on the input length n of the verifier's input. We call $m = m(n)$ the message complexity of the proof system.

It is useful to have a notion of efficiency for the prover as well as for the verifier. The appropriate notion turns out to depend on the "type" of the language L for which the interactive proof is designed.

- $L \in \mathbf{NP}$ and a strategy P_x as above can be implemented in polynomial-time given x and an **NP** witness[21] for x; or

- if $L \in \mathbf{P}$ and a strategy P_x can be implemented in polynomial-time given only x,

then we say that the proof-system has an efficient prover.

Definition 5.11 (Arguments). *An (s, ϵ)-*computationally sound interactive proof *(or* argument*) for a language $\mathcal{L}$ is an interactive proof for $\mathcal{L}$ in which the soundness condition is weakened to:*

- *((s, ϵ)-Computational Soundness): For all $x \in \{0,1\}^n \setminus \mathcal{L}$ and all size-s prover strategies P^*, it holds that $V(x)$ when interacting with P^* accepts with at most ϵ probability.*

[21] The notion of an **NP** witness relies on associating a relation R with the language L; such a relation will usually be implicit from context.

We omit s and ϵ if for all $s = s(n) \leq n^{O(1)}$, the protocol satisfies (s, ϵ)-computational soundness for some $\epsilon = \epsilon(n) \leq \mathrm{negl}(n)$.

Definition 5.12 (Public-Coin)**.** *An interactive proof or argument V is said to be* public-coin *if:*

- *For some $\ell(n) \leq n^{O(1)}$ and every $x \in \{0,1\}^n$, the messages sent by $V(x)$ are i.i.d. uniformly random $\ell(n)$-bit strings.*
- *The final output of $V(x)$ when interacting with a prover P is a fixed polynomial-time computable function of x and the transcript τ of its interaction with P. We denote this output by $V(x, \tau)$.*

For 2-message arguments, we also consider the notion of adaptive soundness, in which a prover may decide what it is trying to prove after seeing the verifier's first message.

Definition 5.13 (Adaptive Soundness)**.** *Let V be a 2-message argument in which the verifier's messages have length $\ell = \ell(n)$ and are* independent *of the statement x.*

*V is said to be (s, ϵ)-*adaptively sound *if for all size-s circuit ensembles P^*, the probability that $V(x, \sigma, \alpha) = 1$ and $x \in \{0,1\}^n \setminus \mathcal{L}$ is at most ϵ when sampling*

$$\begin{aligned} &\sigma \leftarrow \{0,1\}^{\ell(n)} \\ &(x, \alpha) := P^*(1^n, \sigma). \end{aligned}$$

If for all $s = s(n) \leq n^{O(1)}$, V is (s, ϵ)-adaptively sound for some $\epsilon = \epsilon(n) \leq \mathrm{negl}(n)$, then we say simply that V is adaptively sound.

Zero-Knowledge. We recall here the definition of auxiliary-input computational zero-knowledge, referred to henceforth simply as *zero-knowledge*. Our presentation follows [Gol07].

Definition 5.14. *We say that an interactive proof (P, V) for a language L is* zero-knowledge*, if for every (malicious) probabilistic polynomial-time verifier V^* there* *exists a probabilistic polynomial-time simulator* Sim*, such that for every (even inefficient) function* $z = z(x)$*, referred to as the auxiliary input, the following two distribution ensembles are computationally indistinguishable:*

- $\left\{\mathsf{view}_{P,V^*(z(x))}(x)\right\}_{x\in L}$*, where* $\mathsf{view}_{P,V^*}(x)$ *includes the entire view of the verifier* V^* *in the interaction with* P *on common input* x *and auxiliary input* z*; and*
- $\left\{\mathsf{Sim}(x, z(x))\right\}_{x\in L}$.

5.2.2 Cryptographic Primitives and Assumptions

Definition 5.15 (Non-Interactive Statistically Binding Commitments in the CRS Model)**.** *A* non-interactive bit commitment scheme *in the CRS model is a pair of efficient randomized algorithms* $(\mathsf{Setup}, \mathsf{Com})$*, where:*

- $\mathsf{Setup}(1^\lambda)$ *outputs a string* crs*, which we refer to as a common reference string.*
- $\mathsf{Com}(\mathsf{crs}, m; r)$ *takes as input a common reference string* crs *and a message* $m \in \{0,1\}$*; then, using randomness* r*, it outputs a commitment* com*.*

We require the following security properties:

- ***Statistical binding****: With high probability over* $\mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda)$*, there do not exist any two strings* r_0, r_1 *such that* $\mathsf{Com}(\mathsf{crs}, 0; r_0) = \mathsf{Com}(\mathsf{crs}, 1; r_1)$*.*
- ***Computational hiding****: The distribution of* $(\mathsf{crs}, \mathsf{com})$ *when sampling*

$$\begin{aligned}&\mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda)\\&\mathsf{com} \leftarrow \mathsf{Com}(\mathsf{crs}, 0)\end{aligned}$$

is computationally indistinguishable from the distribution when sampling

$$\begin{aligned}&\mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda)\\&\mathsf{com} \leftarrow \mathsf{Com}(\mathsf{crs}, 1).\end{aligned}$$

Given a commitment string com *and common reference string* crs*, we call a valid message-randomness pair* (m, r) *an* opening *for* com*.*

Remark 5.16. *Any public-key encryption scheme* $\mathsf{PKE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *with perfect decryption correctness*[22] *implies a non-interactive commitment scheme in the CRS model: The CRS is a public key* pk*, and a commit to a message* m *is an encryption of* m *under* pk*.*

Moreover, this commitment scheme has the following "trapdoor extractability" property: given the secret key sk *corresponding to* pk *and a (potentially malicious) commitment* com*, one can efficiently compute* m *such that the only possible opening of* com *(if any) is to* m*.*

Learning with Errors (LWE). We next define the learning with errors problem [Reg05].

Definition 5.17. *The* (Decisional) Learning With Errors (LWE) assumption *with parameters* $n = n(\lambda)$*,* $m = m(\lambda)$*,* $q = q(\lambda)$*,* $\chi \in \mathcal{D}(\mathbb{Z}_q)$*, denoted by* $\mathsf{LWE}_{n,m,q,\chi}$*, states that the distribution ensembles* $\{(\mathbf{A}, \mathbf{b})\}_\lambda$ *and* $\{(\mathbf{A}, \mathbf{r})\}_\lambda$ *are computationally indistinguishable, where* $\mathbf{A} \leftarrow \mathbb{Z}_q^{n\times m}$*,* $\mathbf{b}^T = \mathbf{s}^T\mathbf{A} + \mathbf{e}^T \pmod{q}$*,* $\mathbf{s} \leftarrow \mathbb{Z}_q^n$*,* $\mathbf{e} \leftarrow \chi^m$ *and* $\mathbf{r} \leftarrow \mathbb{Z}_q^m$*.*

The subexponential *variant of the* LWE *assumption states that for some* $\epsilon > 0$*, every size-*2^{n^ϵ} *adversary has advantage at most* 2^{-n^ϵ} *in distinguishing these two distributions.*

The subexponential advantage *variant of the* LWE *assumption states that for some* $\epsilon > 0$*, every poly-size adversary has advantage at most* 2^{-n^ϵ} *in distinguishing these two distributions.*

A typical parameter setting for LWE (which suffices for our purposes) is $q = \mathsf{poly}(n)$, $m = \Theta(n \log q)$ and χ defined to be the uniform distribution on $[-B, B] \subseteq \mathbb{Z}_q$ for $B = \mathsf{poly}(\lambda)$ (but significantly smaller than q).

[22] In fact, it is sufficient if for *almost* all key pairs $(\mathsf{pk}, \mathsf{sk})$, it holds for all messages m and randomnesses r that $\mathsf{Dec}\big(\mathsf{sk}, \mathsf{Enc}(\mathsf{pk}, m; r)\big) = m$.

5.2.3 Correlation-Intractable Hash Functions

In this section, we recall the notion of correlation intractable hash functions as introduced by Canetti, Goldreich and Halevi [CGH98].[23] We first give a syntactic definition of keyed hash functions.

Definition 5.18. *A* hash family *is a collection* $\mathcal{H} = \{h_\lambda : \mathcal{I}_\lambda \times X_\lambda \to Y_\lambda\}_{\lambda\in\mathbb{Z}^+}$ *of keyed hash functions such that* $\{\mathcal{I}_\lambda\}$ *is uniformly* $\mathsf{poly}(\lambda)$*-time sampleable and* $\{h_\lambda\}$ *is uniformly* $\mathsf{poly}(\lambda)$*-time evaluable.*

We will also write $\mathcal{H}_\lambda$ *to denote the distribution on functions* $h_\lambda(I, \cdot)$ *obtained by sampling* $I \leftarrow \mathcal{I}_\lambda$*.*

Correlation-intractability is defined as follows.

Definition 5.19 (Correlation-Intractability)**.** *For a hash family* $\mathcal{H} = \{h_\lambda : \mathcal{I}_\lambda \times X_\lambda \to Y_\lambda\}_\lambda$ *and a relation ensemble* $R = \{R_\lambda \subseteq X_\lambda \times Y_\lambda\}$*, the* correlation intractability game $\mathcal{G}^{\mathsf{CI}}_{\mathcal{H},R}$ *is the following game, played by any adversary* $\mathcal{A}$ *against a fixed "challenger"* $\mathcal{C}$*:*

1. *On input* 1^λ*,* $\mathcal{C}$ *samples* $I \leftarrow \mathcal{I}_\lambda$ *and sends* I *to* $\mathcal{A}$*.*
2. $\mathcal{A}$ *sends* $x \in X_\lambda$ *to* $\mathcal{C}$*, and wins the game if* $\big(x, h_\lambda(I, x)\big) \in R_\lambda$*.*

We say that $\mathcal{H}$ *is* $\big(s(\cdot), \epsilon(\cdot)\big)$-correlation intractable for R *if for every size-*$s(\lambda)$ *circuit* $\mathcal{A}$ *and every sufficiently large* λ*, the adversary* $\mathcal{A}$ *wins the correlation intractability game with probability at most* $\epsilon(\lambda)$*.*

If we omit s*, we mean* (s, ϵ)*-security simultaneously for* all $s(\lambda) \le \lambda^{O(1)}$*. If we omit* ϵ*, we mean* (s, ϵ)*-security simultaneously for all* $\epsilon(\lambda) \ge \lambda^{-O(1)}$*.*

Theorem 5.20 ([PS19])**.** *Assume that* $\mathsf{LWE}_{\frac{m}{2\log q},m,q,B}$ *holds for a particular parameter setting* $q = \mathsf{poly}(m), B = q^{\Omega(1)}$*. Then, for every triple of polynomials* $T = T(\lambda), n = n(\lambda), m = m(\lambda)$*, there exists a hash function family* $\mathcal{H} : \{0,1\}^n \to \{0,1\}^{m\log q}$ *that is correlation-intractable for every function ensemble* $f = \{f_\lambda\}_\lambda$ *that is computable in time* $T(\lambda)$*.*

[23] A related security notion was introduced by Okamoto [Oka93] in the context of applying Fiat-Shamir to a particular identification scheme.

5.2.4 The Fiat-Shamir Transform

Definition 5.21. *Let $\Pi = (P, V)$ be a public-coin interactive protocol and denote its messages by $\alpha_1, \beta_1, \ldots, \alpha_r, \beta_r$, where the α_i's are the prover messages and the β_i's are the verifier messages. Suppose that all verifier messages have length ℓ. For a family $\mathcal{H}$ of hash functions mapping $\{0,1\}^* \to \{0,1\}^\ell$, we define $\mathrm{FS}_{\mathcal{H}}[\Pi]$ to be the non-interactive protocol obtained by sampling as a common reference string $h \leftarrow \mathcal{H}$, and replacing each verifier message β_i by $h(x, \alpha_1, \beta_1, \ldots, \alpha_i)$, where x is the main input to the protocol. The verifier for $\mathrm{FS}_{\mathcal{H}}[\Pi]$ accepts if and only if the underlying verifier accepts and all messages β_i were computed correctly.*

In case Π is defined in the CRS model, with CRS σ, then we likewise view $\mathrm{FS}_{\mathcal{H}}[\Pi]$ as a protocol in the CRS model, using the CRS (σ, h).

Definition 5.22. *We say that a hash function family $\mathcal{H}$ is* FS-compatible *with an interactive proof Π for a language $\mathcal{L}$, if the non-interactive protocol $\mathrm{FS}_{\mathcal{H}}[\Pi]$ is an adaptively sound argument for $\mathcal{L}$. We say that $\mathcal{H}$ is* non-adaptively FS-compatible *with Π if $\mathrm{FS}_{\mathcal{H}}[\Pi]$ is a (not necessarily adaptively) sound argument for $\mathcal{L}$.*

We say that $\mathcal{H}$ is FS-compatible (or non-adaptively FS-compatible) with quantitative security *$\mathrm{SubExp}(\lambda)$ (for $\lambda = \lambda(n)$) if in addition there exists $\epsilon > 0$ such that $\mathrm{FS}_{\mathcal{H}}[\Pi]$ is $(2^{\lambda^\epsilon}, 2^{-\lambda^\epsilon})$-computationally sound.*

[DNRS99] established the following negative connection between the existence of FS-compatible hash functions and zero-knowledge.

Theorem 5.23 ([DNRS99])**.** *Let Π be a public-coin interactive proof for a language L. Suppose that there exists an FS-compatible hash function family $\mathcal{H}$ for Π. Then, if Π is zero-knowledge, then $L \in \mathsf{BPP}$.*

The proof of Theorem 5.23 is simple but not exactly in this form in [DNRS99], so we provide a proof for completeness.

Proof Sketch. Suppose that Π is zero-knowledge and consider a cheating verifier V^* that gets as auxiliary input a hash function h and answers each prover message by applying h to the transcript thus far (as in Fiat-Shamir). Since Π is zero-knowledge, there exists a simulator Sim for V^*. Consider a decision procedure D for L that samples a random hash function h, runs $\mathsf{Sim}(x, h)$ (i.e., using h as the auxiliary input) and accepts if any only if (1) the transcript is accepting, and (2) the verifier messages in the transcript are computed correctly (i.e., by applying h).

First observe that if $x \in L$, by the zero-knowledge property the simulated transcript τ is computationally indistinguishible from the real interaction. By completeness, the real interaction produces an accepting transcript and so τ is accepting (and consistent with h) with all but negligible probability. Thus, $D(x)$ accepts with all but negligible probability if $x \in L$.

Next, note that if $x \notin L$, the soundness of $\mathrm{FS}_{\mathcal{H}}(\Pi)$ implies that $D(x)$ accepts with only negligible probability. This is because, given a Fiat-Shamir hash function h, one efficient cheating strategy P^* for $\mathrm{FS}_{\mathcal{H}}(\Pi)$ is to run $\mathsf{Sim}(x, h)$ and send the simulated τ transcript as its message. Therefore, such a transcript can be accepting (and consistent with h) with only negligible probability.

We conclude that D is a BPP algorithm for L. □

5.2.5 Error Correcting Codes and List Recovery

Definition 5.24. *A q-*ary code *is a function $C : \mathcal{M} \to [q]^n$, where n is called the* block length*, $\mathcal{M}$ is called the* message space*, and $[q]$ is called the* alphabet *of C. The* distance *of C is the minimum Hamming distance between $C(m)$ and $C(m')$ for distinct $m, m' \in \mathcal{M}$. If C has distance d, then its* relative distance *is d/n.*

When discussing the asymptotic performance of codes, it makes sense to consider ensembles of codes $\{C_k : \mathcal{M}_k \to [q_k]^{n_k}\}_{k \in \mathbb{Z}^+}$ with varying parameters. We will only consider constructable codes, which are ensembles for which:

- There is an efficiently computable (and invertible) bijection between $\mathcal{M}_k$ and $\left[|\mathcal{M}_k|\right]$, and $|\mathcal{M}_k|$ is computable in time $\mathsf{poly}(k)$.

- q_k, and n_k are computable given 1^k in time $\mathsf{poly}(k)$.

- There is a polynomial-time algorithm E that, given $m \in \mathcal{M}_k$ (represented as an integer in $\big[|\mathcal{M}_k|\big]$), outputs $C_k(m)$.

Definition 5.25 (Concatenated Code [For66]). *Let $C : \mathcal{M} \to [Q]^N$ and $c : [Q] \to [q]^n$ denote codes. The* concatenated code *$C \circ c : \mathcal{M} \to [q]^{Nn}$ is defined by*

$$(C \circ c)(m)_{(i-1)n+j} = c\Big(C(m)_i\Big)_j,$$

for all $m \in \mathcal{M}$, $i \in [N]$, and $j \in [n]$.

Definition 5.26 (List-Recoverable Codes [GI01, GS98]). *An ensemble of codes $\Big\{C_k : \mathcal{M}_k \to [q_k]^{n_k}\Big\}$ is said to be $(\alpha(\cdot), \ell(\cdot), L(\cdot))$-*list recoverable *(for $\alpha : \mathbb{Z}^+ \to (0,1)$ and $\ell, L : \mathbb{Z}^+ \to \mathbb{Z}^+$) if there is a polynomial-time algorithm* Recover *that:*

- *Takes as input $k \in \mathbb{Z}^+$ and explicit descriptions of "constraint" sets $S_1, \ldots, S_{n_k} \subseteq [q_k]$ with each $|S_i| \leq \ell(k)$;*

- *Produces as output a list of at most $L(k)$ messages, containing all $m \in \mathcal{M}_k$ for which $(C_k(m))_i \in S_i$ for at least an $\alpha(k)$ fraction of $i \in [n_k]$.*

The code $\{C_k\}$ is said to be combinatorially *(α, ℓ, L)-*list recoverable *if an arbitrarily inefficient algorithm* Recover *exists with the above functionality. If $\alpha = 1$, we omit it.*

When $\ell = 1$, list recoverability is the same as the more common notion of list decodability.

5.2.6 Concentration Inequalities

Theorem 5.27 (Multiplicative Chernoff). *If $X_1, \ldots, X_n$ are independent $\{0,1\}$-valued random variables with $X \stackrel{\text{def}}{=} \sum_i X_i$ and $\mu \stackrel{\text{def}}{=} \mathbb{E}[X]$, then for all $\delta \geq 0$,*

$$\Pr[X \geq (1+\delta)\mu] \leq \left(\frac{e^\delta}{(1+\delta)^{1+\delta}}\right)^\mu. \tag{5.1}$$

Corollary 5.28. *There is an absolute constant $c > 1$ such that if X and μ are as above, then for any $\tau \geq 3\mu$, we have*

$$\Pr[X \geq \tau] \leq c^{-\tau}.$$

Proof. Follows from viewing τ as $(1+\delta)\mu$ for $\delta \geq 2$ and rewriting Eq. (5.1) as

$$\begin{aligned}\Pr[X \geq (1+\delta)\mu] &\leq \left(\frac{e^{\delta/(1+\delta)}}{1+\delta}\right)^{(1+\delta)\mu} \\ &= \left(\frac{1+\delta}{e^{\delta/(1+\delta)}}\right)^{-\tau} \\ &\leq \left(\frac{3}{e}\right)^{-\tau}. \end{aligned}$$

□

Theorem 5.29 (Additive Chernoff)**.** *If $X_1, \ldots, X_n$ are independent $\{0,1\}$-valued random variables with $X \overset{\mathsf{def}}{=} \sum_i X_i$ and $\mu \overset{\mathsf{def}}{=} \mathbb{E}[X]$, then for all $\epsilon \geq 0$,*

$$\Pr\left[\frac{1}{n}X \geq \mu + \epsilon\right] \leq e^{-2\epsilon^2 n}. \tag{5.2}$$

5.3 Derandomization for Correlation Intractability

In this section, we describe and analyze two derandomization techniques that help achieve correlation intractability for more expressive relation classes. The first technique, described in Section 5.3.1, gives a reduction from CI for (approximate) product relations to CI for functions, based on list-recoverable codes. Next, in Section 5.3.2, we show a generic technique for reducing the alphabet size of product relations using subsampling.

5.3.1 Correlation Intractability via List Recovery

Throughout this section, let $R \subseteq X \times Y^t$ be a binary relation. Our positive result on correlation intractability are for relations with a product structure along with relatively mild sparsity and computational efficiency requirements.

Definition 5.30 (Product Relation)**.** *We say that R is a* product relation *if for every x, the set $R_x = \{y : (x, y) \in R\} \subseteq Y^t$ has a decomposition*

$$R_x = S_1 \times S_2 \times \dots S_t$$

(where $S_1, \dots, S_t$ may depend on x).

We generalize Definition 5.30 to handle (a large fraction of) errors:

Definition 5.31 (Approximate Product Relation)**.** *We say that R is an α-*approximate product relation *if for every x, the set $R_x = \{y \in Y^t : (x, y) \in R\}$ consists exactly of all those $y \in Y^t$ for which*

$$\Big|\{i \in [t] : y_i \in S_i\}\Big| \geq \alpha t.$$

for some sets $S_1, \dots, S_t \subseteq Y$ that may depend on x.

We construct hash functions that are CI for (approximate) product relations satisfying a form of *efficient verifiability.*

Definition 5.32 (Efficient Product Verifiability)**.** *We say that an (α-approximate) relation R is* efficiently product verifiable *if there is a polynomial-size circuit C such that, on every input x with some corresponding sets $(S_1, \dots, S_t)$ (as in Definition 5.31) corresponding to x, it holds that $C(x, y, i) = 1$ if and only if $y \in S_i$.*

Whenever we consider an approximate product relation R, we assume (and, when necessary, provide) a specific decomposition $\Big\{(S_{1,x}, \dots, S_{t,x})\Big\}_{x \in X}$ for R; the decomposition is represented by a circuit deciding membership of y in $S_{i,x}$ given (x, y, i).

The notion of *sparsity* that is most relevant for these relations is simply a bound on the (relative) size of the component sets S_i:

Definition 5.33 (Product Sparsity)**.** *We say that a product (resp., α-approximate product) relation R has* product sparsity *ρ if for every input x, the sets $S_1, \dots, S_t$ as in Definition 5.30 (resp., Definition 5.31) have size at most ρq.*

In order to construct a hash family that is correlation intractable for (approximate) product relations, we simply compose a "base" CI hash function with an appropriate list recoverable code.

Definition 5.34 (Encoded Hash Function)**.** *Let $h : \mathcal{I} \times X \to Z$ be a hash function with index set $\mathcal{I}$, domain X, and codomain Z, and let $\mathcal{C} : Z \to Y^t$ be a (probabilistically) constructable code*[24] *(see Section 5.2.5). We write $\mathcal{C} \circ h$ to denote the hash function $\tilde{h} : \tilde{\mathcal{I}} \times X \to Y^t$, where $\tilde{\mathcal{I}} \overset{\mathsf{def}}{=} \mathcal{I} \times \mathcal{C}$ and $\tilde{h}\Big((i, C), x\Big) \overset{\mathsf{def}}{=} C\Big(h\Big(i, x\Big)\Big)$.*

In order to *analyze* the correlation intractability of encoded hash functions, we introduce a kind of *derandomization* for (approximate) product relations R, which will help us achieve correlation intractability for R.

Specifically, if $R \subseteq X \times Y^t$ is a product (or approximate product) relation and $C : Z \to Y^t$ is a code, we define

$$\tilde{R}_C = \Big\{(x, z) : \Big(x, C(z)\Big) \in R\Big\}.$$

The following lemma then holds syntactically.

Lemma 5.35. *If $\mathcal{C}$ is a (probabilistically) constructable code ensemble, $R = \{R_\lambda\}$ is a relation (ensemble) and if $\mathcal{H}$ is a hash family that is correlation intractable for $\tilde{R}_C$, then $\mathcal{C} \circ \mathcal{H}$ as in Definition 5.34 is a hash family that is correlation intractable for R.*

Proof. Suppose that $\mathcal{C} \circ \mathcal{H}$ is *not* CI for R; then, there exists an efficient adversary $\mathcal{A}(h)$ that on input $h \leftarrow \mathcal{H}$, outputs an x such that $(x, C(h(x)) \in R$ with non-negligible probability. By the definition of $\tilde{R}_C$, we know that $(x, C(h(x)) \in R$ implies that $(x, h(x)) \in \tilde{R}_C$, so this contradicts the CI of $\mathcal{H}$ with respect to $\tilde{R}_C$. □

Theorem 5.36. *Let T be an arbitrary time bound; then, define $\mathcal{R} = \mathcal{R}_{\alpha,\epsilon,T}$ to be the class of all time-T verifiable α-approximate product relations $R \subseteq X \times Y^t$ with product sparsity $1 - \epsilon$. Moreover, suppose that*

[24] We omit the parameterization of h (respectively $\mathcal{C}$) by a security parameter (respectively the message length) for simplicity.

- *$C : Z \to Y^t$ is a code that is $(\alpha, (1-\epsilon)q, L)$ list-recoverable in $\mathsf{poly}(L)$-time, with $L = \mathsf{poly}(n, q)$;* and

- *The hash family $\mathcal{H}$ is (quantitatively $\frac{\mathrm{negl}(\lambda)}{T'}$-) correlation intractable for all functions that are computable within some sufficiently large time bound $T' = \mathsf{poly}(T, t, |Y|)$.*

Then, $C \circ \mathcal{H}$ is correlation intractable for all relations in $\mathcal{R}$. In particular, when $T, |Y|, t$ are all fixed polynomials in a security parameter λ, then if $\mathcal{H}$ is CI for functions *computable in $\mathsf{poly}(\lambda)$ time, then $C \circ \mathcal{H}$ is CI for $\mathcal{R}$.*

Proof. Lemma 5.35 tells us that for any time-T verifiable (approximate) product relation R, the hash family $\mathcal{H}'$ is CI for R as long as $\mathcal{H}$ is CI for the derandomized relation $\tilde{R}$ above.

We now claim that subject to the hypotheses above, $\tilde{R}$ is *efficiently enumerable* in the sense of [CCH+19]: there is an efficient (meaning $\mathsf{poly}(T, t, |Y|)$) algorithm that, given x, enumerates all $z \in Z$ such that $(x, z) \in \tilde{R}_C$. Indeed, this is possible via the following procedure:

- First, construct the sets $S_1, \ldots, S_t$ given $x = (x_1, \ldots, x_t)$; this can be done in time $t \cdot T \cdot |Y|$.

- Then, evaluate $\mathsf{Recover}(S_1, \ldots, S_t)$. By the correctness of list-recovery, this produces (with high probability) a poly-size list of all $z \in Z$ for which $(x, z) \in \tilde{R}_C$.

The runtime of this entire enumeration procedure is a fixed polynomial $\mathsf{poly}(T, t, |Y|)$. Finally, we recall that in [CCH+19] (see [CLW18] Section 3.1), it was noted that if $\mathcal{H}$ is ϵ-CI for time-T' computable functions, then it is $\frac{\epsilon}{T'}$-CI for time-T' enumerable relations (and in particular $\tilde{R}$); thus, we conclude that $\mathcal{H}'$ is CI for R with the claimed quantitative parameters. □

5.3.2 Handling Large Alphabets via Subsampling

While Theorem 5.36 could plausibly apply to product relations in $X \times Y^t$ with $|Y| = \lambda^{\omega(1)}$, our instantiations (Theorems 5.42 and 5.56) can only directly handle alphabets of size $|Y| = \mathsf{poly}(\lambda)$; this is because we employ list-recovery algorithms that take as input (uncompressed) lists of size $|Y|^{\Omega(1)}$ (and we also explicitly assume that $t \geq |Y|^{\Omega(1)}$ in our code constructions).

However, we can achieve correlation intractability even for large values of $|Y|$ — assuming we have sparsity $\rho \leq 1 - \frac{1}{\mathsf{poly}(\lambda)}$. We do so by first *subsampling* a random sub-alphabet $\tilde{Y} \subseteq Y$ and *restricting* the relation R to this sub-alphabet. That is, given a relation $R \subseteq X \times Y^t$ and alphabet $\tilde{Y} \subseteq Y$, we define

$$R_{\tilde{Y}} = R \cap \left(X \times \tilde{Y}^t\right) \tag{5.3}$$

We note that:

- If membership in a set S_i can be verified in time T, then membership in $S_i \cap \tilde{Y}$ can be verified in time $T + |\tilde{Y}| \log |Y|$.
- A hash function that is CI for $R_{\tilde{Y}}$ is *also* CI for R when viewed as a hash function with output space $\tilde{Y}^t \subseteq Y^t$.

Moreover, we note that a sufficiently large (random) subset of Y preserves the sparsity of the S_i under intersection.

Lemma 5.37. *Suppose that $R \subseteq X \times Y^t$ is an α-approximate product relation with product sparsity ρ. For some $\epsilon > 0$ and $\lambda \in \mathbb{Z}^+$, let $\tilde{Y} \subseteq Y$ be a uniformly random subset of size $q \geq \frac{\log|X| + \log t + \lambda}{\epsilon^2}$, i.e. $\tilde{Y}$ is sampled uniformly at random from $\binom{Y}{q}$.*

Then $R_{\tilde{Y}}$ as defined in Eq. (5.3) is an α-approximate product relation that, with probability $1 - 2^{-\lambda}$ over the choice of $\tilde{Y} \leftarrow \binom{Y}{q}$, has product sparsity $\leq \rho + \epsilon$.

Proof. This follows from union bounding over $|X| \cdot t$ subsets $S_{ij} \subseteq Y$ (depending on the relation and indexed by $i \in X$, $j \in [t]$), each of size at most $\rho|Y|$. For each such S_{ij}, when $\tilde{Y}$ is sampled as above, it holds with probability at least $1 - (t \cdot |X| \cdot 2^{\lambda})^{-2}$

that the intersection $S_{ij} \cap \tilde{Y}$ has size at most $(\rho+\epsilon) \cdot |\tilde{Y}|$. This follows from a standard Chernoff bound (Theorem 5.29). □

We conclude that sub-sampling gives a reduction from CI over large alphabets to CI over polynomial-size alphabets.

Corollary 5.38. *Let $R \subseteq X \times Y^t$ be an α-approximate product relation with product sparsity ρ, and let $q = q(\lambda)$ be an integer such that $q \geq \frac{\log|X|+\log t+\lambda}{\epsilon^2}$, where λ is a computational security parameter.*

Suppose that for each $\tilde{Y} \in \binom{Y}{q}$, $\mathcal{H}_{\tilde{Y}}$ is a family of hash functions mapping $X \to \tilde{Y}^t$ that is CI for α-approximate product relations with product sparsity $\rho + \epsilon$. Then the hash family $\mathcal{H}$, where a random element of $\mathcal{H}$ is sampled as $h \leftarrow \mathcal{H}_{\tilde{Y}}$ for uniformly random $\tilde{Y} \leftarrow \binom{Y}{q}$, is CI for R.

Corollary 5.38 will be used in Section 5.5 and Section 5.6 to obtain CI hash functions with large output alphabets, which in turn yields Fiat-Shamir instantiations for parallel repetitions of interactive proofs with large verifier challenge spaces.

5.4 Basic List Recovery Bounds

In this section, we recall and rephrase some facts about the list-recoverability of three objects from the coding-theory literature: Parvaresh-Vardy codes [PV05,GR08], random codes (as analyzed by [GI01]), and generic code concatenation [For66]. These bounds will be used in Section 5.5 and Section 5.6 to build new codes that combine with Theorem 5.36 in different ways.

We begin with a description of what is achieved by Parvaresh-Vardy codes.

Theorem 5.39 (Parvaresh-Vardy codes [PV05, GR08])**.** *There is an explicit code*

$$C : [q]^k \to [q^s]^q,$$

parameterized by integers $s, k, q \in \mathbb{Z}^+$ (with q a power of two) such that for every $\alpha \in [0, 1]$, the code is (efficiently) (α, ℓ, L)-list recoverable in time $\mathsf{poly}\Big((2s)^s, q, \ell\Big)$ as *long as*

$$\ell < \left(\frac{\alpha}{s+1}\right)^{s+1} \cdot \frac{q^s}{k^s}$$

and

$$L > c \cdot (2s)^s \cdot \frac{q\ell}{k}$$

for some absolute constant c.

C can be evaluated in time less than the above bound on the time required to list recover.

We also need bounds on the list-recoverability of random codes. List-recovery bounds for random (and flavors of pseudorandom) codes were stated (but not proved) in [GI01]; for completeness we prove here the results that we use. We first give the fully parameterized result and then specialize to parameter regimes of interest.

Theorem 5.40. *There exists a constant $c > 0$ such that for any q, Q, α, ℓ, and L (all of which are functions of n), a random function $f : [Q] \to [q]^n$ is combinatorially (α, ℓ, L)-list recoverable with probability $1 - 2^{-\Omega(L)}$ as long as*

$$L \geq c \cdot \left(Q \cdot \rho + \ell \cdot n \cdot \log\left(\frac{q}{\ell}\right)\right), \tag{5.4}$$

where the parameter ρ is

$$\rho \overset{\mathsf{def}}{=} \Pr\left[\mathsf{Binom}\Big(n, \ell/q\Big) \geq \alpha n\right].$$

This list recovery can be done (by brute force) in time $O(Q \cdot n \cdot \ell \cdot \log q)$. Evaluation of f can be done in time $O(Q \cdot n \cdot \log q)$.

Theorem 5.40 follows by a straightforward application of the probabilistic method, details follow.

Proof. Let f be a random function mapping $[Q] \to [q]^n$. We want to show that with high probability, for all sets $S_1, \ldots, S_n \subseteq [q]$ of size ℓ, the size of the set $f^{-1}(B)$ is at most L, where by B we denote

$$B \overset{\mathsf{def}}{=} \Big\{ z \in [q]^n : |i \in [n] : z_i \in S_i| \geq \alpha n \Big\}.$$

We analyze this by union bounding over $\binom{q}{\ell}^n \leq \left(\frac{q \cdot e}{\ell}\right)^{\ell n}$ events corresponding to the possible choices of $S_1, \ldots, S_n$.

To analyze an individual one of these events, we note that for fixed sets $S_1, \ldots, S_n$, the random variable $|f^{-1}(B)|$ follows a binomial distribution $\mathsf{Binom}\,(Q, \tilde{\rho})$, for

$$\begin{aligned} \tilde{\rho} &\overset{\mathsf{def}}{=} \Pr_{z \leftarrow [q]^n} \Big[z \in B \Big] \\ &= \Pr_{z \leftarrow [q]^n} \Big[\Big| i \in [n] : z_i \in S_i \Big| \geq \alpha n \Big] \\ &\leq \Pr \Big[\mathsf{Binom}\Big(n, \ell/q\Big) \geq \alpha n \Big] \\ &= \rho. \end{aligned}$$

A multiplicative Chernoff bound (Corollary 5.28) implies that for some constant $c_0 > 0$,

$$\Pr \Big[\Big| f^{-1}(B) \Big| > L \Big] < c_0^{-L},$$

provided that $L > 3\tilde{\rho} \cdot Q$.

Then, the union bound gives the desired conclusion about f as long as

$$c_0^{-L} \cdot \left(\frac{q \cdot e}{\ell} \right)^{\ell n} \leq 2^{-\Omega(L)},$$

which holds if $L \geq c_1 \cdot \ell n \cdot \log\left(\frac{q}{\ell}\right)$ for some absolute constant $c_1 > 0$. Combining these two conditions on L yields Theorem 5.40. □

We also make use of the (known) fact that concatenated codes inherit list recoverability from their constituent parts.

Lemma 5.41. *Suppose that*

- $C : \mathcal{M} \to [Q]^N$ *is an* $(\frac{\alpha - \beta}{1 - \beta}, \ell', L)$*-list recoverable code and*

- *$c : [Q] \to [q]^n$ is a (β, ℓ, ℓ')-list recoverable code*

for $1 \geq \alpha > \beta > 0$ and $\ell, L \in \mathbb{Z}^+$. Then $C \circ c$ is (α, ℓ, L)-list recoverable. Moreover, if list-recovery for C can be computed in time T and list-recovery for c can be computed in time t, then list-recovery for $C \circ c$ can be computed in time $T + n \cdot t$.

In the special case of errorless list recovery ($\alpha = 1$), it suffices for C to be (ℓ', L)-list recoverable and c to be (ℓ', ℓ)-list recoverable to imply that $C \circ c$ is (ℓ, L)-list recoverable.

Proof. Let $\{S_{i,j}\}_{i\in[N],j\in[n]}$ be subsets of $[q]$ of size at most ℓ. We want to bound the size of the set

$$S \overset{\mathsf{def}}{=} \{m \in \mathcal{M} : (C \circ c)(m)_{i,j} \in S_{i,j} \text{ for at least } \alpha n N \text{ choices of } (i, j)\} .$$

To do this, we note that by Markov's inequality, for any $m \in S$ with $(C \circ c)(m) = z_{1,1}, \ldots, z_{N,n}$, we have

$$\left| \{i \in [N] : z_{i,j} \in S_{i,j} \text{ for at least } \beta \cdot n \text{ choices of } j \in [n]\} \right| \geq \frac{\alpha - \beta}{1 - \beta} N.$$

Therefore, the (β, ℓ', ℓ)-list recoverability of c implies that there exist lists $L^{(1)}, \ldots, L^{(N)} \subseteq [Q]$ of size at most ℓ' such that for all such m, $C(m)_i \in L^{(i)}$ for at least $\frac{\alpha-\beta}{1-\beta}N$ choices of i. By the $(\frac{\alpha-\beta}{1-\beta}, \ell, L)$-list recoverability of C_1, there are at most L such messages m.

Moreover, the collection of such messages can be recovered in time $T + N \cdot t$ via the following algorithm:

- Given the collection of sets $\{S_{i,j}\}$, compute the lists $L^{(1)}, \ldots, L^{(N)}$ defined by these sets (with respect to c) in (total) time $N \cdot t$.

- Then, run the list recovery algorithm for C on $L^{(1)}, \ldots, L^{(N)}$ to obtain the final list.

Finally, in the case of errorless list recovery, we have by assumption that *all* symbols $z_{i,j}$ of each block have to lie in the appropriate $S_{i,j}$, so the claim follows. □

5.5 Fiat-Shamir for Commit-And-Open Protocols

In this section, we obtain our positive results for Fiat-Shamir by applying Theorem 5.36.

In Section 5.5.1, give a CI instantiation for product relations (Theorem 5.42). To prove this theorem, we give a randomized construction of codes that are (ℓ, L)-list recoverable for large values of ℓ; the codes are obtained by concatenating Parvaresh-Vardy codes [PV05] with a random code. We carefully choose the parameters of the two codes to optimize the block-length of the concatenation. We augment Theorem 5.42 with alphabet reduction (Corollary 5.38) to handle larger alphabets.

In Sections 5.5.2 and 5.5.3, we state and prove our Fiat-Shamir instantiations. We give a general result for (parallel repetitions of) 3-message protocols with efficiently verifiable bad challenges, and then focus on commit-and-open protocols (Definition 5.51) as a special case. Finally, in Section 5.5.4, we state our negative results on parallel repeated zero knowledge, which are obtained by invoking [DNRS99].

5.5.1 Correlation Intractability for Efficiently Verifiable Product Relations

Our main result in this section is a construction of correlation intractable hash families for product relations over polynomial-size alphabets.

Theorem 5.42. *Let $R = R_\lambda \subseteq X_\lambda \times Y_\lambda^{t_\lambda}$ be an ensemble of product relations that are time-$T(\lambda)$ product-verifiable as in Definition 5.32 with product sparsity at most ρ, where $|Y_\lambda|$, $\log |X_\lambda|$, $T(\lambda)$, and t_λ are all upper bounded by $\lambda^{O(1)}$ and $t_\lambda \geq \lambda / \log(1/\rho)$.*

Then there exists a hash family $\mathcal{H} = \{\mathcal{H}_\lambda : X_\lambda \to Y_\lambda^{t_\lambda}\}_{\lambda \in \mathbb{Z}^+}$ that is correlation intractable for R under the LWE *assumption. Moreover, $\mathcal{H}$ depends only on (X, Y, ρ, t, T) (and is otherwise independent of R) and can be evaluated in time*

$\mathsf{poly}(\log|X|, |Y|, t, T)$.

Several remarks follow on the efficiency properties of Theorem 5.42:

- The dependence of the evaluation time on $\mathcal{H}$ on $|Y|$ can be reduced if R's product decomposition can be computed explicitly in time $\ll |Y| \cdot T$ (which is the generic bound for a time T-product verifiable relation). This can apply in situations where ρ is very small. Alternatively, all dependencies on $|Y|$ can be generically reduced via alphabet reduction (see Theorem 5.46).

- If we write $\rho = 1 - \epsilon$, it suffices to have $t_\lambda \geq \lambda/\epsilon$ (which approaches $\lambda/\log(1/\rho)$ for small ϵ).

- With our usual "polynomial hardness" notions of security—that is, requiring that any $\mathsf{poly}(\lambda)$-size adversary cannot win correlation intractability games with probability $\lambda^{-\Omega(1)}$—it is equivalent (by a standard scaling argument) to replace this requirement by the seemingly weaker requirement that $t_\lambda \geq \lambda^\delta/\log(1/\rho)$ for any arbitrarily small constant $\delta > 0$.

- Under a sub-exponential variant of LWE, the requirement that $t_\lambda \geq \lambda/\log(1/\rho)$ can be weakened to $t_\lambda \geq \log^c \lambda/\log(1/\rho)$ for a large enough constant c, while still retaining standard polynomial security in the correlation intractability of the resulting hash family.

- On the other hand, correlation intractability against larger adversaries (or smaller success probabilities) is also achievable by increasing t_λ. For example, assuming sub-exponential LWE, it is possible to achieve security against size-2^λ adversaries by requiring $t_\lambda \geq \lambda^c/\log(1/\rho)$ for a sufficiently large constant c.

To prove Theorem 5.42, we first construct a family of list-recoverable codes for our parameter regime of interest. We start with the following proposition, which follows immediately from Theorem 5.39 and gives list recoverable codes in the errorless case (i.e., $\alpha = 1$), with polynomial input list sizes, output list size, alphabet and block length.

Proposition 5.43. *For all constants c and all $\ell = \ell(k) \leq k^c$, there exists a constructable ensemble of codes*

$$C = \left\{ C_k : \{0,1\}^k \to [Q_k]^{O(k^2)} \right\}_{k \in \mathbb{Z}^+}$$

for some $\ell \leq Q_k \leq O(\ell^4)$ such that C_k is $\Big(\ell, O(k\ell)\Big)$-list recoverable in time $\mathsf{poly}(k, \ell, c^c)$.

Proof. Suppose we are given a polynomially bounded function $\ell(\cdot)$. We obtain such an ensemble from Theorem 5.39 by setting:

- $\alpha = 1$;
- $s = 2\log_k(\ell)$, which is bounded by a constant depending on $\ell(\cdot)$; and
- q is the smallest power of two that is at least k^2,

which results in $Q_k \leq (2k^2)^s \leq O(\ell^4)$. □

We remark that it is also possible to set $q = O(k^{1+\epsilon})$ for an arbitrarily small constant $\epsilon > 0$, which would result in a slightly better bound for the block length of C, but no qualitatively new applications for Fiat-Shamir.

We additionally use the following bound on the list recoverability of random functions (without errors), which follows from Theorem 5.40

Proposition 5.44. *For all $\ell = \ell(k)$, $q = q(k)$, $L = L(k)$, and $Q = Q(k)$ satisfying $\ell < q$ and $L \geq \ell \cdot \omega(\log Q)$, setting $n = n(k) = \left\lceil \frac{\log Q}{\log(q/\ell)} \right\rceil$, a random function $f : [Q] \to [q]^n$ is combinatorially $\Big(\ell, L\Big)$-list recoverable with probability $1 - 2^{-\Omega(L)}$.*

Proof. We apply Theorem 5.40 with $\alpha = 1$ and $n = \left\lceil \frac{\log Q}{\log(q/\ell)} \right\rceil$, which ensures that $\rho \overset{\mathsf{def}}{=} (\ell/q)^n$ is at most $1/Q$. Because $n \cdot \log(q/\ell)$ is $O(\log Q)$, we have that $L \geq \omega\Big(Q \cdot \rho + \ell \cdot n \cdot \log \frac{q}{\ell}\Big)$, from which it follows by Theorem 5.40 that f is combinatorially (ℓ, L)-list recoverable with probability $1 - 2^{-\Omega(L)}$. □

Concatenating the codes of Propositions 5.43 and 5.44 yields codes with list recoverability parameters that are useful for our applications.

Lemma 5.45. *For all $\ell = \ell(k)$, $q = q(k)$ with $\ell < q \leq k^{O(1)}$, there exists a probabilistically constructable ensemble of codes*

$$\left\{C_k : \{0,1\}^k \to [q]^{O\left(\frac{k^2 \log(k)}{\log(q/\ell)}\right)}\right\}$$

such that each C_k is $\Big(\ell, O(k^2\ell)\Big)$-list recoverable in time $k^{O(1)}$ with all but $2^{-\Omega(k)}$ probability.

More precisely, the running time is a fixed polynomial in k, ℓ, $\log_k(\ell)^{\log_k(\ell)}$, and $\frac{1}{\log(q/\ell)}$.

Proof. Consider any choice of $\ell = \ell(k)$ and $q = q(k)$ as above. Then by Proposition 5.43, there is some $Q = Q_k \leq \Big(k\ell\Big)^4$ and a constructable ensemble of codes $C' = \{C'_k : \{0,1\}^k \to [Q]^{O(k^2)}\}$ that is $\Big(k\ell, O(k^2\ell)\Big)$-list recoverable in time $k^{O(1)}$. For this Q, Proposition 5.44 guarantees that with $n = n(k) = \left\lceil \frac{\log Q}{\log(q/\ell)} \right\rceil$, a random function $f = f_k : [Q] \to [q]^n$ is combinatorially $(\ell, k\ell)$-list recoverable with probability $1 - 2^{-\Omega(k\ell)}$. Such an f is sampleable in time $(k\ell)^4 \cdot n \log(q)$ because $Q_k \leq (k\ell)^4$. Similarly, the brute-force $(\ell, k\ell)$-list recovery algorithm for f runs in time $(k\ell)^4 \cdot \ell n \log(q)$. Concatenating C' with f yields the desired ensemble:

$$\left\{(C'_k \circ f_k) : \{0,1\}^k \to [q]^{k^2 \cdot n}\right\}_{k \in \mathbb{Z}^+}$$

is $\Big(\ell(k), O(k^2\ell)\Big)$-list recoverable in $k^{O(1)}$ time by Lemma 5.41. □

Proof of Theorem 5.42. We are finally ready to prove our main theorem. We compose (a random instance of) the code from Lemma 5.45 with the CI hash family of Theorem 5.20. Theorem 5.36 then implies that the composition yields a good correlation-intractable hash family for the claimed relations.

The Large Alphabet Case. Finally, we combine Theorem 5.42 with Corollary 5.38 to obtain the following result on CI for product relations over *large alphabets*. We specialize this result to the case $\rho = 1 - \epsilon$ for convenience.[25]

Theorem 5.46. *Let $R \subseteq X_\lambda \times Y_\lambda^{t_\lambda}$ be a product relation that is time-T product-verifiable (where $\log |X|, T, t = \mathsf{poly}(\lambda)$) with product sparsity at most $1 - \epsilon$ for $\epsilon \geq \lambda^{-O(1)}$.*

Then, if $t \geq \lambda/\epsilon$, there exists a hash family $\mathcal{H} = \{\mathcal{H}_\lambda : X_\lambda \to Y_\lambda^{t_\lambda}\}_{\lambda \in \mathbb{Z}^+}$ that is correlation intractable for R under the LWE *assumption. Moreover, $\mathcal{H}$ depends only on $(X_\lambda, Y_\lambda, T_\lambda, t_\lambda, \epsilon)$ and can be evaluated in time $\mathsf{poly}(\log |X|, t, T)$.*

5.5.2 Fiat-Shamir for Trapdoor 3-Message Protocols

We now describe a general Fiat-Shamir instantiation for 3-message public coin interactive proofs with *trapdoor decidable bad challenges*, defined below. This notion is a generalization of (instance-dependent) trapdoor Σ-protocols as defined in [CCH+19].

Definition 5.47 (Bad-Challenge Relation)**.** *Let Π denote a 3-message public coin interactive proof system for a language $\mathcal{L}$ in the (possibly empty) CRS model. We define the* bad challenge relation *$R^{(\Pi,\mathsf{crs})}$ for Π (with a fixed CRS* crs*) to be*

$$R^{(\Pi,\mathsf{crs})} = \Big\{(x|\alpha,\, \beta) : x \notin \mathcal{L} \text{ and } \exists \gamma : V(\mathsf{crs}, x, \alpha, \beta, \gamma) = 1\Big\}.$$

For an instance $x \notin \mathcal{L}$, we define the non-adaptive bad challenge relation *$R^{(\Pi,\mathsf{crs},x)}$ to be*

$$R^{(\Pi,\mathsf{crs},x)} = \Big\{(\alpha, \beta) : \exists \gamma : V(\mathsf{crs}, x, \alpha, \beta, \gamma) = 1\Big\}.$$

Definition 5.48 (Trapdoor Decidable Bad Challenges)**.** *We say that a 3-message public-coin proof system Π for a language $\mathcal{L}$ in the CRS model has* (time-T) trapdoor decidable bad challenges *if there exist*

- *An efficient algorithm $\mathsf{TrapGen}(1^\lambda)$ that outputs a pair $(\mathsf{crs}, \mathsf{td})$;*
- *A sparse binary relation $R^{(\mathsf{td})}$; and*

[25] The approximations incurred by this specialization cost at most a factor of $O(\log(\lambda))$ in the number of repetitions our techniques can achieve for *exact* product relations.

- *An algorithm* $\mathsf{BadChallengeTest}(\mathsf{td}, x, \alpha, \beta)$ *that takes as input the trapdoor* td*, the instance* x*, a first message* α*, and a second message (or challenge)* β*,*

satisfying the following properties:

- *When sampling* $(\mathsf{crs}, \mathsf{td}) \leftarrow \mathsf{TrapGen}(1^\lambda)$*, the distribution of* crs *is statistically indistinguishable from that of an honestly generated CRS.*

- $R^{(\mathsf{td})}$ *contains the bad-challenge relation* $R^{(\Pi,\mathsf{crs})}$ *(Definition 5.47).*

- $\mathsf{BadChallengeTest}(\mathsf{td}, x, \alpha, \beta)$ *runs in time* T *and outputs* 1 *if and only if* $(x|\alpha, \beta) \in R^{(\mathsf{crs})}$*.*

Definition 5.49. *We say that* Π *has* (time-T) *instance-dependent* trapdoor decidable bad challenges *if it satisfies Definition 5.48 with the following modifications:*

- $\mathsf{TrapGen}(1^\lambda, w)$ *also takes as input non-uniform advice* w *about the instance* x*; and*

- $\mathsf{BadChallengeTest}$ *and* $R^{(\mathsf{td},x)}$ *are defined with respect to the* non-adaptive *bad challenge relation* $R^{(\Pi,\mathsf{crs},x)}$ *instead of with respect to* $R^{(\Pi,\mathsf{crs})}$*.*

- *CRS indistinguishability is only required to be computational.*

By applying Theorems 5.42 and 5.46, we obtain Fiat-Shamir instantiations for 3-message proof systems Π with (instance-dependent) trapdoor decidable bad challenges.

Theorem 5.50. *Suppose that* Π *is a* 3*-message public coin proof system that has time-*T *trapdoor decidable bad challenges, such that the relation* $R^{(\mathsf{td})}$ *in Definition 5.48 has sparsity at most* $1-\epsilon$ *for* $\epsilon \geq \lambda^{-O(1)}$*. Then, for any* $t \geq \lambda/\epsilon$*, there exists a hash family* $\mathcal{H}$ *that is FS-compatible with* $\Pi^{(t)}$ *(guaranteeing* adaptive soundness*).*

Similarly, if Π *has time-*T *instance-dependent trapdoor decidable bad challenges and each* $R^{(\mathsf{td},x)}$ *has sparsity at most* $1-\epsilon$*, then there exists an* $\mathcal{H}$ *that is FS-compatible with* $\Pi^{(t)}$ *(guaranteeing selective soundness).*

In both cases, $\mathcal{H}$ *can be evaluated in time* $\mathsf{poly}(T, t, \lambda)$*.*

Proof Sketch. By (statistical) CRS indistinguishability, we may assume that crs is sampled as $(\mathsf{crs}, \mathsf{td}) \leftarrow \mathsf{TrapGen}(1^\lambda)$ in the (adaptive) soundness security game. If Π is repeated t times in parallel, then following [CCH+19], we know that $\mathcal{H}$ is FS-compatible with $\Pi^{(t)}$ if it is CI for the relation

$$R = \left\{ \left(x|\alpha_1|\dots|\alpha_t,\ \beta_1|\dots|\beta_t\right) : \exists\gamma : V\left(x, \alpha_i, \beta_i, \gamma_i\right) = 1 \text{ for all } i. \right\}$$

By the definition of $\mathsf{BadChallengeTest}$, R is *contained* in the relation

$$R' = \left\{ \left(x|\alpha_1|\dots|\alpha_t,\ \beta_1|\dots|\beta_t\right) : \mathsf{BadChallengeTest}(\mathsf{td}, x, \alpha_i, \beta_i) = 1 \text{ for all } i \right\},$$

By assumption on $\mathsf{TrapGen}$, R' is a time T verifiable product relation with product sparsity at most $1-\epsilon$. Thus, the theorem follows from Theorem 5.42. The proof for the non-adaptive case is analogous. □

5.5.3 Commit and Open Protocols

A commit-and-open protocol is a ubiquitous type of protocol that always has trapdoor-decidable bad challenges when the commitment scheme is instantiated using public-key encryption.

Definition 5.51 (3-Message Commit-and-Open Protocol)**.** *An interactive proof system for a language $\mathcal{L}$ is said to be* commit-and-open *if it is defined relative to a statistically binding commitment oracle* Com *and has the following structure.*

1. *The verifier takes as an input a string $x \in \{0,1\}^n$.*

2. *The prover sends a message α consisting of a string of commitments $(\mathsf{com}_i)_{i\in[M]}$ for some $M = M(n) \leq n^{O(1)}$.*

3. *The verifier V sends a random challenge $\beta \leftarrow [q]$ for some $q = q(n)$.*

4. *The prover sends a message γ containing openings of the commitments $(\mathsf{com}_i)_{i\in S_\beta}$, where $S_\beta \subseteq [M]$ is a set that is efficiently computable from β. We denote the i^{th} such opening by (b_i, d_i).*

5. *The verifier checks that for each $i \in S_\beta$, (b_i, d_i) is a valid opening of com_i (and otherwise rejects). If so, the verifier accepts if some predicate $V\Big(x, \alpha, \beta, (b_i)_{i=1}^{|S_\beta|}\Big) = 1$. In particular, this predicate ignores the d_i's.*

The important attributes of Definition 5.51 (that distinguish commit-and-open protocols from arbitrary 3-message protocols) are that the third message *only* consists of openings, and that the verifier rejects incorrect openings and otherwise ignores the decommitments d_i. Therefore, by instantiating Com using a public-key encryption scheme, the PKE decryption key sk allows for efficient verification of whether a challenge β is "bad" for a pair (x, α), because the bits (b_i) for a valid decommitment can be *extracted* from α using sk.

Lemma 5.52. *Let Π denote a 3-message commit-and-open protocol. Then, if Com is instantiated using a public-key encryption scheme as in Remark 5.16, then Π has time-T trapdoor decidable bad challenges, where T is equal to the runtime of $V(\cdot)$ plus a fixed polynomial in the security parameter.*

Proof. We give Π the syntax of a protocol with trapdoor decidable bad challenges as follows:

- $\mathsf{TrapGen}(1^\lambda)$ is defined to sample $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^\lambda)$ and output $(\mathsf{crs} = \mathsf{pk}, \mathsf{td} = \mathsf{sk})$.
- The relation $R^{(\mathsf{td})}$ is defined as

$$R^{(\mathsf{td})} = \Big\{(x|\alpha,\ \beta) : V(x, \alpha, (b_i)_{i=1}^{|S_\beta|}) = 1,\ \text{ for } b_i = \mathsf{Dec}(\mathsf{sk}, \mathsf{com}_i)\Big\}.$$

- The algorithm $\mathsf{BadChallengeTest}(\mathsf{td}, x, \alpha, \beta)$ parses $\alpha = (\mathsf{com}_i)$, computes $b_i = \mathsf{Dec}(\mathsf{sk}, \mathsf{com}_i)$ for all $i \in S_\beta$, and computes $V(x, \alpha, (b_i)_{i=1}^{|S_\beta|})$.

Correctness follows immediately from the decryption correctness property of PKE. □

Given Lemma 5.52, Theorem 5.46 implies that all commit-and-open protocols have sound Fiat-Shamir instantiations when sufficiently repeated in parallel.

Theorem 5.53. *Assume that* LWE *holds, and let* Π *be any 3-message commit-and-open interactive proof with soundness error* $1-\epsilon$*, where* $\epsilon \geq \lambda^{-O(1)}$ *for a computational security parameter* λ*. Let the commitment scheme in* Π *be instantiated using public-key encryption as in Remark 5.16.*

Then for any $t = t(\lambda) \geq \lambda/\epsilon$*, there is a hash family* $\mathcal{H}$ *that is Fiat-Shamir compatible with* Π^t *as in Definition 5.22. The hash functions in* $\mathcal{H}$ *are evaluable in time* $T \cdot \mathsf{poly}(\lambda)$*, where* $T = T(n)$ *is the running time of the verifier for* Π*, and* n *is the length of an input for* Π*.*

Proof. This follows immediately from Theorem 5.50 and Lemma 5.52. □

5.5.4 Zero Knowledge is Not Preserved by Parallel Repetition

Finally, we invoke [DNRS99] to conclude that parallel repetition of commit-and-open protocols (such as GMW) does not preserve zero-knowledge.

Theorem 5.54. *Assume that* LWE *holds. Then, there exists a commitment scheme* C *such that for every 3-message commit-and-open proof-system* Π *in the commitment oracle model (as per Definition 5.51) for a language* $L \notin \mathbf{BPP}$ *with soundness error* $1-\epsilon$*, it holds that* $(\Pi_C)^t$ *is not zero-knowledge, where* Π_C *denotes the instantiation of the commitment oracle in* Π *by* C *and* $t = \lambda/\epsilon$ *for a security parameter* λ*.*

Proof. Fix C to be the public-key encryption based commitment-scheme of Remark 5.16. Let Π be a 3-message commit-and-open proof-system with soundness error $1-\epsilon$ (in the commitment oracle model). Denote by Π_C the instantiation of Π using C in place of the commitment oracle. By Theorem 5.53, there exists a hash function family $\mathcal{H}$ such that $\mathrm{FS}_{\mathcal{H}}[(\Pi_C)^t]$ is computationally sound, where $t = \lambda/\epsilon$. In other words, $\mathcal{H}$ is FS-compatible with $(\Pi_C)^t$.

Thus, by Theorem 5.23, and using the assumption that $L \notin \mathbf{BPP}$, we have that $(\Pi_C)^t$ is not zero-knowledge. □

Remark 5.55. *Assuming the subexponential variant of LWE, the number of repetitions t in Theorem 5.54 can be reduced to $\frac{\log^c \lambda}{\epsilon}$ for some constant $c > 1$ (in fact, under the strongest plausible LWE assumption c can even be $1+\delta(n)$ for some $\delta(n) = o(1)$). This still leaves open the somewhat bizarre possibility that for some very specific values of t (e.g., $t = \log(\lambda) \cdot \log^\star(\lambda)$), the parallel repeated protocol Π^t is both sound and zero knowledge.*

*In fact, it is known that this sort of gap is difficult to avoid: [BLV03] show that for any HVZK commit-and-open protocol Π with poly-size challenge space, if Circuit-*SAT *has a $2^{o(n)}$ time algorithm, then some $\omega(1)$-parallel repetition of Π* is *zero knowledge. Thus, resolving this gap implies an exponential lower bound for Circuit-*SAT.

5.6 Fiat-Shamir for Round-By-Round Sound Protocols

In this section, we extend the results of Section 5.5 to the setting of Fiat-Shamir for *multi-round* protocols. We achieve this in three steps.

- In Section 5.6.1, we construct a (probabilistic) code with efficient list recovery *in the presence of errors.*

- We then combine this code with Lemma 5.35 to obtain a CI hash family for efficiently verifiable approximate product relations.

- In Section 5.6.2, we apply our new CI hash family to instantiate FS for a family of round-by-round sound interactive proofs. There are three variants of this result, depending on the precise efficiency requirement imposed on the interactive proof. In particular, we achieve FS instantiations for a larger class of protocols by making use of *lossy* correlation intractability [JKKZ21].

5.6.1 CI for Efficiently Verifiable Approximate Product Relations

Our main result in this section is a construction of correlation intractable hash families for *approximate* product relations (see Definition 5.31) over polynomial-size alphabets.

Theorem 5.56. *Let $R \subseteq X \times Y^t$ be a time-T verifiable α-approximate product relation with product sparsity at most $\rho < \alpha$.*

Set $\lambda = t \cdot (\alpha - \rho)^3$. Then, assuming that all $\mathsf{poly}(T, \lambda)$-time adversaries solve LWE[26] *with probability at most ϵ, there is a hash family $\mathcal{H} = \mathcal{H}_n$ that is $\Big(T + \mathsf{poly}(\lambda), \epsilon \cdot \mathsf{poly}(\lambda)\Big)$-correlation intractable for R.*

Moreover, $\mathcal{H}$ depends only on (X, Y, T, t, α) (and not otherwise on R).

Remark 5.57. *By pre-composing our hash family $\mathcal{H}$ with a lossy trapdoor function, we also obtain a hash family $\mathcal{H}'$ satisfying* lossy correlation intractability *[JKKZ21] for the same class $\mathcal{R}$ of relations.*

To prove Theorem 5.56, we first construct a family of list-recoverable codes in the presence of errors. We begin by describing the salient list recovery (with errors) properties of Parvaresh-Vardy codes, which follow as a corollary of Theorem 5.39.

Proposition 5.58. *For every $\alpha = \alpha(k) \geq k^{-O(1)}$ and every $\ell = \ell(k) \leq k^{O(1)}$, there exists $Q(k) \leq k^{O(1)}$ and a constructable ensemble of codes*

$$\Big\{ C_k : \{0,1\}^k \to [Q(k)]^{O(k^2/\alpha)} \Big\}_{k \in \mathbb{Z}^+}$$

that is $(\alpha, \ell, O(k\ell/\alpha))$-list recoverable in time $k^{O(1)}$, where the exponent depends on all previous parameters.

More precisely, $Q(k)$ is bounded by $\Big(2k^2/\alpha(k)\Big)^{2\log_k(\ell/\alpha)}$ and the list recovery algorithm's running time is a fixed polynomial in k, ℓ, $1/\alpha$, and $\log_k(\ell/\alpha)^{\log_k(\ell/\alpha)}$.

Proof. We obtain such an ensemble from Theorem 5.39 by setting:

[26] Specifically, we need to assume the hardness of $\mathsf{LWE}_{n,m,q,\chi}$ for $n = O(\frac{\lambda}{\log \lambda})$, $m = 2n \log q$, $q = \lambda^{O(1)}$, and χ the uniform distribution on $[-B, B]$ for some $B = \lambda^{\Omega(1)}$, as in Definition 5.17.

- q to be the smallest power of two that is at least $k^2/\alpha(k)$ (which is $k^{O(1)}$ because $\alpha(k)$ is $k^{-O(1)}$);
- s to be a large enough constant (depending on α) so that $\left(\frac{\alpha}{s+1}\right)^{s+1} \cdot \frac{q^s}{k^s} > \ell$ for all sufficiently large k. Specifically, one should set s to be the smallest integer that is at least $\log_k(\ell/\alpha)$. □

Next, we describe a corollary of Theorem 5.40, which (similarly to Proposition 5.44) focuses on asymptotics, this time for list recovery with errors.

Proposition 5.59. *For all $q = q(k)$, $\ell = \ell(k)$, $Q = Q(k)$, and $\alpha = \alpha(k)$, $\ell < q$, and $\alpha > \frac{\ell}{q}$, there exists $L = L(k) \leq O\left(\frac{\ell \cdot \log(Q) \cdot \log \frac{q}{\ell}}{(\alpha - \ell/q)^2}\right)$ such that a random function*

$$f : [Q] \to [q]^{\frac{\log Q}{2(\alpha - \ell/q)^2}}$$

is combinatorially (α, ℓ, L)-list recoverable with all but $2^{-\Omega(L)}$ probability.

Proof. Given q, ℓ, L, and α as above, define $n = n(k) = \frac{\log Q}{2(\alpha - \ell/q)^2}$. This n is big enough that by the additive Chernoff bound (Theorem 5.29), we have

$$\rho \overset{\mathsf{def}}{=} \Pr[\mathsf{Binom}(n, \ell/q) \geq \alpha n] \leq 1/Q.$$

Then setting $L = c \cdot \left(Q \cdot \rho + \ell \cdot n \cdot \log \frac{q}{\ell}\right)$ for a large enough constant c and applying Theorem 5.40 implies the corollary. □

By concatenating the two codes above (with carefully chosen parameters), we obtain codes with list recoverability parameters that are useful for our applications.

Lemma 5.60. *For every $\ell = \ell(k)$, $q = q(k)$, $\alpha = \alpha(k)$ with $\ell < q \leq k^{O(1)}$ and $\alpha \geq \ell/q + k^{-O(1)}$, there is a probabilistically constructable ensemble of codes*

$$\left\{C_k : \{0,1\}^k \to [q]^{O\left(\frac{k^2 \log k}{(\alpha - \ell/q)^3}\right)}\right\}_{k \in \mathbb{Z}^+}$$

that is $(\alpha, \ell, k^{O(1)})$-list recoverable in time $k^{O(1)}$ with all but $2^{-\Omega(k)}$ probability.

More precisely,[27] *the running time of the list recovery algorithm is a fixed polynomial in* $k, \ell, \frac{1}{\alpha-\ell/q}$*, and* Q^* *for*

$$\log_k(Q^*) \leq 2\log_k\left(\frac{8\ell k\log(q/\ell)}{\left(\alpha-\frac{\ell}{q}\right)^3}\right)\log_k\left(\frac{4k^2}{\alpha-\frac{\ell}{q}}\right) = O(1).$$

Proof. Suppose we are given ℓ, q, and α as above. Let $\beta \overset{\mathsf{def}}{=} \frac{1}{2}(\alpha+\frac{\ell}{q})$. Proposition 5.59 guarantees that for all $Q = Q(k) \leq k^{O(1)}$, there is some $L_Q = L_Q(k) \leq k^{O(1)}$ such that with $n(k) = \frac{\log Q}{2(\beta-\ell/q)^2}$, a random function $f_Q : [Q] \to [q]^n$ is combinatorially (β, ℓ, L_Q)-list recoverable with all but $2^{-\Omega(L_Q)}$ probability. More precisely, this $L_Q(k)$ satisfies $L_Q(k) \leq O\left(\frac{\ell\cdot\log(Q)\cdot\log\frac{q}{\ell}}{(\beta-\ell/q)^2}\right)$, which is $O\left(\frac{\ell\cdot\log k\cdot\log\frac{q}{\ell}}{(\beta-\ell/q)^2}\right)$ because we required that $Q \leq k^{O(1)}$. For the same reason, the brute force list recovery algorithm for such an f_Q is efficient (running in time $O(Q\cdot n\cdot\log q) = k^{O(1)}$).

Let $L^\star = L^\star(k)$ satisfy $\omega\left(\frac{\ell\cdot\log(Q)\cdot\log\frac{q}{\ell}}{(\beta-\ell/q)^2}\right) \leq L^\star \leq k^{O(1)}$ (for instance, set $L^\star = \frac{\ell\cdot k\cdot\log\frac{q}{\ell}}{(\beta-\ell/q)^2}$). Setting $\tilde{\alpha} = \frac{\alpha-\beta}{1-\beta}$, Proposition 5.58 guarantees the existence of $Q^\star = Q^\star(k) \leq k^{O(1)}$ such that there is a constructable ensemble of codes

$$C = \left\{C_k : \{0,1\}^k \to [Q^*(k)]^{O(k^2/\tilde{\alpha})}\right\}$$

that is $\left(\tilde{\alpha}, L^\star, O\left(\frac{kL^\star}{\tilde{\alpha}}\right)\right)$-list recoverable. More precisely, Proposition 5.58 gives $Q^\star(k) \leq (2k^2/\tilde{\alpha})^{2\log_k(L^\star/\tilde{\alpha})}$. Also note that

$$\tilde{\alpha} = \frac{\alpha-\beta}{1-\beta} \geq \alpha-\beta = \frac{1}{2}\cdot(\alpha-\ell/q) \geq k^{-O(1)}$$

and

$$\beta - \frac{\ell}{q} = \frac{1}{2}(\alpha-\frac{\ell}{q}).$$

Choosing $f_Q = f_{Q^\star}$ from above, we conclude that the concatenation

$$C\circ f : \{0,1\}^k \to [q]^{O(nk^2/\tilde{\alpha})}.$$

[27] We write down this explicit expression because it determines the runtime of the Fiat-Shamir hash functions in Theorem 5.68.

satisfies our desired properties by Lemma 5.41. □

By plugging the (randomized) code from Lemma 5.60 and the hash family of Theorem 5.20 into Theorem 5.36, we obtain Theorem 5.56.

5.6.2 Applications to Fiat-Shamir for Round-by-Round Sound Protocols

Following [CCH+18, CCH+19], we consider the notion of round-by-round soundness to capture a form of soundness for interactive proofs of greater than 3 messages that is compatible with the notion of correlation intractability.

Definition 5.61 (Round-by-Round Soundness, [CCH+18,CCH+19])**.** *Let* $\Pi = (P, V)$ *be a* $2r + 1$*-message public coin interactive proof system for a language* L*. We say that* Π *has* round-by-round soundness error $\delta(\cdot)$ *(or is* δ-RBR sound*) if there is a deterministic (not necessarily efficiently computable) function* State*, which takes as input an instance* x *and a transcript prefix* τ *and outputs either* acc *or* rej *such that the following holds:*

1. *If* $x \notin L$*, then* $\mathsf{State}(x, \emptyset) = \mathsf{rej}$*, where* $\emptyset$ *denotes the empty transcript.*
2. *For every input* x *and partial transcript* $\tau = \tau_i$*, if* $\mathsf{State}(x, \tau) = \mathsf{rej}$*, then for every potential prover message* α_{i+1}*, it holds that*

$$\Pr_{\beta_{i+1}}\Big[\mathsf{State}\Big(x, \tau|\alpha_{i+1}|\beta_{i+1}\Big) = \mathsf{acc}\Big] \leq \delta(n).$$

3. *For any* full *transcript* τ *(i.e., consisting of* $2r+1$ *messages), if* $\mathsf{State}(x, \tau) = \mathsf{rej}$ *then* $V(x, \tau) = 0$*.*

We say that Π *is* round-by-round sound *if it has round-by-round soundness error* δ *for some* $\delta(n) = \mathrm{negl}(n)$*.*

By a union bound, a proof system with round-by-round soundness error δ has standard soundness error at most $r \cdot \delta$.

Canetti *et al.* [CCH+18] related the soundness of Fiat-Shamir, when applied to a round-by-round sound protocol, to the correlation intractability of the hash function $\mathcal{H}$.

Theorem 5.62 ([CCH+18, Theorem 5.8])**.** *Suppose that $\Pi = (P, V)$ is a $2r+1$-message public-coin interactive proof for a language L with perfect completeness and round-by-round soundness with state function* State*. Let X_n denote the set of partial transcripts (including the input and all messages sent) and let Y_n denote the set of verifier messages when Π is executed on an input of length n.*

Finally, define the relation ensemble $R = R_{\mathsf{State}}$ as follows:

$$R^{(n)}_{\mathsf{State}} \overset{\mathsf{def}}{=} \left\{ \Big(\big(x, \tau|\alpha\big), \beta \Big) : \begin{array}{c} x \in \{0,1\}^n, \\ \mathsf{State}(x,\tau) = \mathsf{rej}, \ \textit{and} \\ \mathsf{State}(x, \tau|\alpha|\beta) = \mathsf{acc} \end{array} \right\}.$$

If a hash family $\mathcal{H} = \{\mathcal{H}_n : X_n \to Y_n\}$ is correlation intractable for R, then the non-interactive protocol $\Pi_{\mathrm{FS},\mathcal{H}}$ is an adaptively sound argument system for L.

In this work, we consider protocols Π with round-by-round soundness error $\rho < \frac{1}{r}$. We then consider applying the Fiat-Shamir transform to a parallel repetition Π^t (for sufficiently large t). To analyze this, we must also analyze how parallel repetition works for round-by-round sound protocols.[28]

Definition 5.63 (Threshold State Function)**.** *Let Π denote a $2r+1$-message public-coin interactive proof system with round-by-round soundness δ and corresponding state function* State*. We then define the* threshold state function $\mathsf{State}^{(t)}$ *defined on the t-fold parallel repetition Π^t: decomposing a (partial) transcript of $\Pi^{(t)}$ as a tuple $(\tau_1, \ldots, \tau_t)$ (where each τ_i is a partial transcript for Π), we define*

$$\mathsf{State}^{(t)}(x, \tau_1, \ldots, \tau_t) = \mathsf{rej} \iff \Big| \{ i \in [t] : \mathsf{State}(x, \tau_i) = \mathsf{rej} \} \Big| \geq 1 + \frac{r-j}{r} \cdot (t-1),$$

[28] In [CCH+18, CCH+19], it is noted that sufficient parallel repetition of *any* public-coin interactive proof results in a round-by-round sound protocol, but this transformation results in a rather complex State function; we want a transformation that roughly *preserves* the State function of the starting protocol (which we assume to satisfy some form of RBR soundness).

where j is the number of verifier messages in each τ_i.

Lemma 5.64. *If Π is a protocol as in Definition 5.63, then $\mathsf{State}^{(t)}$ gives $\Pi^{(t)}$ the structure of a round-by-round sound proof system with RBR soundness error bounded by*

$$\delta^{(t)} := \exp\left(-2\left(\frac{t-1}{r\cdot t} - \delta\right)^2 t\right),$$

provided that $\frac{t-1}{r\cdot t} > \delta$.

Proof. This follows from the fact that for any partial transcript $(x, \tau_1, \dots, \tau_t)$, if $\mathsf{State}^{(t)}(x, \tau_1, \dots, \tau_t) = \mathsf{rej}$ but $\mathsf{State}^{(t)}(x, \tau_1|\alpha_{1,j+1}|\beta_{1,j+1}, \dots, \tau_t|\alpha_{t,j+1}|\beta_{t,j+1}) = \mathsf{acc}$, then at least $\frac{t-1}{r}$ "slots" of τ changed from rej to acc according to State. Thus, for any τ such that $\mathsf{State}^{(t)}(x, \tau) = \mathsf{rej}$ and any $\alpha = (\alpha_{1,j+1}, \dots, \alpha_{t,j+1})$, the probability over β that $\mathsf{State}^{(t)}(x, \tau|\alpha|\beta) = \mathsf{acc}$ is at most the probability that at least $\frac{t-1}{r}$ out of t i.i.d. Bernoulli events with mean δ occur. By a Chernoff bound, this happens with probability at most $\delta^{(t)}$, as desired. □

The proof of Lemma 5.64 in fact shows that the "bad challenge relation" for $(\Pi^{(t)}, \mathsf{State}^{(t)})$ is an *α-approximate product relation* with product sparsity δ, where $\alpha = \frac{t-1}{r\cdot t}$. Therefore, if the relation $R_{\mathsf{State}^{(t)}}$ is efficiently product-verifiable (or, equivalently, the relation R_{State} is efficiently verifiable), we can apply Theorem 5.56 to obtain a sound Fiat-Shamir instantiation for the protocol $\Pi^{(t)}$, provided that t is large enough.

Notions of Bad Challenge Efficient Decidability

In this section, let Π be a $2r+1$-message (public-coin) interactive proof system for a language $\mathcal{L}$.

Definition 5.65 (Trapdoor Decidable Bad Challenges)**.** *We say that public-coin interactive proof Π for a language $\mathcal{L}$ in the CRS model has* round-by-round soundness error δ with time-T trapdoor decidable bad challenges *if there exist*

- *An efficient algorithm $\mathsf{TrapGen}(1^\lambda)$ that outputs a pair $(\mathsf{crs}, \mathsf{td})$;*

- *A δ-sparse binary relation $R^{(\mathsf{td})}$; and*
- *An algorithm* $\mathsf{BadChallengeTest}(\mathsf{td}, x, j, \tau_{j-1}|\alpha_j, \beta_j)$ *that takes as input the trapdoor* td*, the instance x, a transcript prefix $\tau_{j-1}|\alpha_j$ (consisting of j prover messages and $j-1$ verifier messages), and a verifier message β_j,*

satisfying the following properties:

- *When sampling* $(\mathsf{crs}, \mathsf{td}) \leftarrow \mathsf{TrapGen}(1^\lambda)$*, the distribution of* crs *is statistically indistinguishable from that of an honestly generated CRS.*
- $R^{(\mathsf{td})}$ *contains the bad-challenge relation R_{State} (Definition 5.47).*
- $\mathsf{BadChallengeTest}(\mathsf{td}, x, j, \tau_{j-1}|\alpha_j, \beta_j)$ *runs in time T and outputs 1 if and only if $(x|\tau_{j-1}|\alpha_j,\ \beta_j) \in R^{(\mathsf{State})}$.*

Definition 5.65 is a strict generalization of Definition 5.48 and captures the multiround protocols for which we can instantiate Fiat-Shamir based on polynomial hardness of appropriately chosen CI hash families. As in Section 5.5.3, a similar definition captures *non-adaptively sound* Fiat-Shamir instantiations.

Definition 5.66. *We say that Π has* round-by-round soundness error δ with time-T *instance-dependent* trapdoor decidable bad challenges *if it satisfies Definition 5.65 with the following modifications:*

- $\mathsf{TrapGen}(1^\lambda, w)$ *also takes as input non-uniform advice w about the instance x; and*
- $\mathsf{BadChallengeTest}$ *and $R^{(\mathsf{td},x)}$ are defined with respect to the* non-adaptive *bad challenge relation*

$$R_{\mathsf{State},x} = \left\{ (\tau|\alpha,\ \beta) : \begin{array}{c} \mathsf{State}(x,\tau) = \mathsf{rej},\ \ \textit{and} \\ \mathsf{State}(x, \tau|\alpha|\beta) = \mathsf{acc} \end{array} \right\}.$$

  instead of with respect to R_{State}.
- *CRS indistinguishability is only required to be computational.*

Finally, we give a third definition further generalizing the previous two in a way that captures the Sumcheck and GKR protocols with *succinct* bad challenge testing. However, this variant requires stronger assumptions on the CI hash compiler.

Definition 5.67. *We say that* Π *has* round-by-round soundness error δ with time-T *prefix-dependent* trapdoor decidable bad challenges *if it satisfies Definition 5.65 with the following modifications:*

- $\mathsf{TrapGen}(1^\lambda, z_{\beta^*})$ *also takes as input non-uniform advice* $z = f(x, \beta^*)$ *about the instance* x and *a string* $\beta^* = (\beta_1^*, \ldots, \beta_r^*)$ *consisting of (fixed) verifier messages.*

- $\mathsf{BadChallengeTest}$ *and* $R^{(\mathsf{td},x)}$ *are defined with respect to the* non-adaptive *bad challenge relation* $R^{(\mathsf{State},x)}$ *instead of with respect to* R_{State}. Moreover*, "correctness" is relaxed to the following set containment: for all rounds* j *and strings* β^**, when sampling* $(\mathsf{crs}, \mathsf{td}) \leftarrow \mathsf{TrapGen}(1^\lambda, f(x, \beta^*))$,

$$R^{(\mathsf{td},x)} \supseteq \Big\{(\tau|\alpha_j,\ \beta_j) \in R^{(\mathsf{State},x)} : (\beta_1, \ldots, \beta_{j-1}) = (\beta_1^*, \ldots, \beta_{j-1}^*)\Big\}.$$

- *CRS indistinguishability is only required to be computational.*

Putting Everything Together

Given our efficient bad challenge notions from Section 5.6.2 and our CI hash family from Section 5.6.1, we are ready to state our Fiat-Shamir result for round-by-round sound protocols.

Theorem 5.68. *Let* Π *be a* $2r+1$*-message (public-coin) interactive proof system for a language* $\mathcal{L}$ *in which the verifier's messages are uniformly random on* $[q]$ *for some* $q \in \mathbb{Z}^+$ *and prover messages are bit strings of length* $a = a(n)$*. Let* $\delta = \delta(n) \in (0, 1)$ *and* $\lambda = \lambda(n) \in \mathbb{Z}^+$ *be functions, and define*

$$t = \frac{\lambda}{(\frac{1}{2r} - \delta)^3}.$$

Then, there exists $T_{\mathsf{Dec}} = T_{\mathsf{Dec}}(n)$ that is a polynomial in λ, δq, $\frac{1}{\frac{1}{2r}-\delta}$, and $Q^\star$, where

$$\log_\lambda Q^\star = \log_\lambda\left(\frac{8\delta q\log(1/\delta)}{\left(\frac{1}{2r}-\delta\right)^3}\right)\log_\lambda\left(\frac{\lambda^2}{\frac{1}{2r}-\delta}\right),$$

such that:

- *If Π has round-by-round soundness error δ with time-T trapdoor decidable bad challenges, then assuming the hardness of LWE there is a hash family $\mathcal{H}$ that is adaptively FS-compatible with Π^t as in Definition 5.22.*
- *If Π has round-by-round soundness error δ with time-T* instance-dependent *trapdoor decidable bad challenges, then assuming the hardness of LWE there is a hash family $\mathcal{H}$ that is* non-adaptively *FS-compatible with Π^t.*
- *If Π has round-by-round soundness error δ with time-T* prefix-dependent *trapdoor decidable bad challenges, then under the* subexponential advantage *variant of the LWE assumption, there is a hash family $\mathcal{H}$ that is non-adaptively FS-compatible with Π^t.*

Moreover,

- *Assuming subexponential hardness for LWE, the first two results extend to also give FS-compatibility with* SubExp(λ) *quantitative security.*
- *These hash families depend only on $(a(\cdot), q(\cdot), \delta(\cdot), T(\cdot), \lambda(\cdot), r(\cdot))$ and otherwise do not depend on Π.*
- *Hash function evaluation can be done in time that is $O\Big((qT + T_{\mathsf{Dec}}) \cdot \mathsf{poly}(\lambda)\Big)$. The qT term can also be replaced by the amount of time required to enumerate bad challenges for Π.*

Example Application: Fiat-Shamir for GKR We now sketch how, assuming subexponential LWE, Theorem 5.68 allows us to soundly apply the Fiat-Shamir transform to the doubly-efficient public-coin interactive proof of Goldwasser, Kalai, and

Rothblum [GKR08]. This interactive proof, which we refer to as GKR, is applicable to (log-space uniform) bounded-depth computations.

We will fix some family of (log-space uniform) circuits with depth $d = d(n)$ and size $s = s(n)$. GKR is additionally parameterized by a finite field of order $q = q(n)$. The best efficiency (in our case) is achieved for q a power of two, which yields the following parameters:

- The round complexity is $O(d \cdot \log n)$;
- The prover runs in time $\mathsf{poly}(s, \log q)$;
- The verifier runs in time $n \cdot \mathsf{poly}(d, \log s, \log q)$;
- The proof system has round-by-round soundness error $\delta = \delta(n)$ with time-T ($= T(n)$) prefix-dependent trapdoor decidable bad challenges, where $\delta = O(\frac{\log n}{q})$ and $T = \mathsf{poly}(\log n, \log s, \log q)$.

  In particular, the number of bad verifier challenges at any round is $\ell = O(\log n)$

When applying the Fiat-Shamir transform to GKR, we would like to preserve the feature that the verifier's running time is much less than (and ideally polylogarithmic in) the time required to evaluate the circuit. Specifically, we would like the Fiat-Shamir hash functions to be evaluable in time $n \cdot \mathsf{poly}(d, \log q, \log s, \log n)$. This was done by [JKKZ21] for very large q, i.e. $q > (d\ell)^{\kappa^{1/\epsilon}}$ for a computational security parameter κ. Here we focus on the other extreme of parameter settings, where q is small (say $\mathsf{polylog}(n)$), and soundness is amplified by parallel repetition.

To accomplish this, when applying Theorem 5.68 we set $\lambda = (d \log n) \cdot \kappa^{1/\epsilon}$, where ϵ denotes the exponent of our subexponential LWE assumption. Applying Theorem 5.68, we bound the runtime of the verifier as follows. First, note that

$$\frac{1}{2r} - \delta \geq \frac{1}{4r} = \frac{1}{4d \log(n)},$$

and so $\log_\lambda(Q^\star) = O(A \cdot B)$ for

$$A = \log_\lambda(8d\ell \log(\lambda) \log(n)) = O(1)$$

and

$$B = \log_{\lambda}(\lambda^2 d \log(n))) = O(1).$$

Therefore, $Q^\star = \mathsf{poly}(\lambda) = \mathsf{poly}(d, \kappa)$ and so verification runs in time $n \cdot \mathsf{poly}(d, \kappa, \log s, q, \log n)$, which is $n \cdot \mathsf{poly}(d, \kappa, \log s, \log n)$ by the assumption that q is $\mathsf{polylog}(n)$.

Finally, we note that:

- Because Theorem 5.68 gives us sub-exponential security in κ, if our goal is to achieve $\mathsf{poly}(n)$ security (i.e., $\mathrm{negl}(n)$ soundness error against $\mathsf{poly}(n)$ size provers), we can set $\kappa = \mathsf{polylog}(n)$. Then, the hash function evaluation time (and hence the verifier running time) will be $\tilde{O}(n)$.

- By using a root-finding algorithm (see [CCH+19, JKKZ21]) instead of a root verification algorithm (as used in Theorem 5.68 above) in our CI analysis, we can reduce the verifier runtime dependence on q to $\mathsf{poly}(\ell, \log q)$ (instead of $\mathsf{poly}(q)$), enabling us to handle all field sizes (not just polylogarithmic).

- At the expense of a larger number of repetitions (incurring a multiplicative overhead of q), we could replace our Parvaresh-Vardy based code with a concatenation of a Reed-Solomon code with a random code for a faster running time of the hash function (i.e. some fixed polynomial in (λ, d) for all choices of λ, d instead of explicitly requiring $\lambda \geq d \log(n)$).

Chapter 6

2-Message Publicly Verifiable WI from (Subexponential) LWE

6.1 Introduction

In this note, we consider the question of constructing 2-message witness indistinguishable (WI) arguments for NP that are *publicly verifiable*; that is, the argument system consists of a single verifier message followed by a single prover message, and anyone can verify a proof given only the transcript.

In a seminal work, Dwork and Naor [DN00] showed that such argument systems can be constructed given any non-interactive zero knowledge (NIZK) proof system in the common random string model; given the state-of-the-art on NIZK, this yields constructions assuming the hardness of factoring [FLS90] as well as under falsifiable assumptions on bilinear maps [CHK03, GOS06].

In recent work, Canetti et al. [CCH$^+$19] and Peikert and Shiehian [PS19] gave constructions of NIZK argument systems from *lattice assumptions*[1]; however, the [DN00] transformation cannot be directly applied to these constructions in order to obtain 2-message WI arguments. The issue is that both of these works construct NIZKs that are either (1) statistically sound, but requiring a structured common reference

[1] [CCH$^+$19] gave a construction from a circular-secure variant of the learning with errors (LWE) assumption, while [PS19] weakened the assumption to plain LWE.

string, or (2) using a uniformly random CRS, but only satisfying soundness against computationally bounded provers. On the other hand, the [DN00] transformation crucially assumes that the underlying NIZK satisfies statistical soundness and uses a uniformly random CRS.

In this work, we show that a slight modification of the [DN00] transformation can be applied to the [CCH$^+$19, PS19] NIZKs in order to obtain 2-message publicly verifiable WI arguments for NP. Unlike the [DN00] construction, we rely on *complexity leveraging* in order to prove soundness of the 2-message argument system, so we must rely on the subexponential hardness of LWE in order to prove security. As a result, we obtain the following theorem.

Theorem 6.1. *Assuming the subexponential hardness of LWE, there exist two-message publicly verifiable WI arguments for NP.*

We construct two variants of such an argument system: in one variant, soundness is *adaptive* (that is, soundness holds even when the cheating prover is allowed to choose the false statement that he wants to prove), while in the other, the protocol is *public-coin* (that is, the verifier message is a uniformly random string). Both variants are "delayed-input" protocols – meaning that the verifier message does not depend on the instance x – so in either variant, the verifier message can be reused across many executions (even for different statements).

While our construction can be seen as a new variant of the [DN00] transformation from NIZKs to 2-message arguments, we choose to present the construction as a compiler from (sufficiently structured) "trapdoor Σ-protocols" [CCH$^+$19] to 2-message arguments, combining a special-purpose instantiation of the Fiat-Shamir heuristic with a [DN00]-like transformation. More specifically, we give a construction combining dual Regev encryption with the correlation intractable hash families of [CCH$^+$19, PS19].

6.1.1 Concurrent Work

In concurrent and independent works, Badrinarayan et al. [BFJ+20] and Jain and Jin [JJ19] note essentially the same construction of 2-message WI arguments from LWE. Moreover, they give an exciting extension of the result that yields a 2-message (publicly verifiable) WI argument system satisfying *statistical witness indistinguishability.* Such argument systems were not previously known under any standard cryptographic assumption, and we do not give such a construction in this note.

6.2 Preliminaries

We say that a function $\mu(\lambda)$ is *negligible* if $\mu(\lambda) = O(\lambda^{-c})$ for every constant c, and that two distribution ensembles $X = \{X_\lambda\}$ and $Y = \{Y_\lambda\}$ are computationally indistinguishable ($X \approx_c Y$) if for all polynomial-sized circuit ensembles $\{\mathcal{A}_\lambda\}$,

$$\left| \Pr\left[\mathcal{A}_\lambda(X_\lambda) = 1\right] - \Pr\left[\mathcal{A}_n(Y_\lambda) = 1\right] \right| = \mathrm{negl}(\lambda).$$

6.2.1 Witness Indistinguishable Arguments

Definition 6.2. *A* witness indistinguishable arugment system Π *for an* **NP** *relation* R *consists of ppt interactive algorithms* (P, V) *with the following syntax.*

- $P(x, w)$ *is an interactive algorithm that takes as input an instance* x *and witness* w *that* $(x, w) \in R$.
- $V(x)$ *is an interactive algorithm that takes as input an instance* x*. At the end of an interaction, it outputs a bit* b*. If* $b = 1$*, we say that* V accepts*, and otherwise we say that* V rejects.

The proof system Π *must satisfy the following requirements for every polynomial function* $n = n(\lambda)$*. Recall that* $\mathcal{L}(R)$ *denotes the language* $\{x : \exists w \text{ s.t. } (x, w) \in R\}$ *and* R_n *denotes the set* $R \cap (\{0,1\}^n \times \{0,1\}^*)$.

- **Completeness.** *For every* $(x, w) \in R$, *it holds with probability* 1 *that* V *accepts at the end of an interaction* $\langle P(x, w), V(x)\rangle$.

- **Soundness.** *For every* $\left\{x_n \in \{0,1\}^n \setminus \mathcal{L}(R)\right\}$ *and every polynomial size* $P^* = \{P^*_\lambda\}$, *there is a negligible function* ν *such that* V *accepts with probability* $\nu(\lambda)$ *at the end of an interaction* $\langle P^*(x), V(x)\rangle$.

- **Witness Indistinguishability.** *For every ppt (malicious) verifier* V^* *and every ensemble* $\left\{(x_n, (w_{0,n}, w_{1,n}), z_n) : (x_n, w_{0,n}), (x_n, w_{1,n}) \in R_n\right\}$, *the distribution ensembles*

  $$\mathsf{view}_{V^*}\langle P(x, w_0), V^*(x, w_0, w_1, z)\rangle$$

  and

  $$\mathsf{view}_{V^*}\langle P(x, w_1), V^*(x, w_0, w_1, z)\rangle$$

  are computationally indistinguishable.

In the work, we focus on obtaining two message WI arguments for **NP**. A (two message) WI argument system can also satisfy various stronger properties. We list some important variants below.

- **Publicly Verifiable**: A WI argument system is publicly verifiable if the verifier's accept/reject algorithm is an efficiently computable function of the transcript (independent of the verifier's internal state).

- **Public Coin**: A WI argument system is *public coin* if all (honest) verifier messages are uniformly random strings (sampled independently of the protocol so far). Note that any public coin protocol is publicly verifiable.

- **Delayed Input**: A *two-message* WI argument system is *delayed input* if the (honestly sampled) verifier message does not depend on the instance x.

- **Adaptive Soundness**: A *two-message, delayed-input* protocol Π is $\mathsf{adaptively\ sound}$ if for every polynomial size algorithm $P^* = \{P^*_\lambda\}$, there is a negligible function ν such that for all λ,

$$\Pr_{\substack{\mathsf{crs}\leftarrow V(x)\\(x,\pi):=P^*_\lambda(\mathsf{crs})}}[x \notin \mathcal{L}(R) \wedge V(\mathsf{crs}, x, \pi) = 1] \le \nu(\lambda).$$

6.3 Correlation Intractable Hash Families

In this section, we recall the notion of correlation intractability [CGH98], specialization to "efficiently-searchable relations" [CCH+19], and LWE-based instantiation [PS19].

Definition 6.3. *For a pair of efficiently computable functions $(n(\cdot), m(\cdot))$, a* hash family *with input length n and output length m is a collection $\mathcal{H} = \{h_\lambda : \{0,1\}^{s(\lambda)} \times \{0,1\}^{n(\lambda)} \to \{0,1\}^{m(\lambda)}\}_{\lambda\in\mathbb{N}}$ of keyed hash functions, along with a pair of p.p.t. algorithms:*

- $\mathcal{H}.\mathsf{Gen}(1^\lambda)$ *outputs a hash key $k \in \{0,1\}^{s(\lambda)}$.*
- $\mathcal{H}.\mathsf{Hash}(k, x)$ *computes the function $h_\lambda(k, x)$. We may use the notation $h(k, x)$ to denote hash evaluation when the hash family is clear from context.*

We cay that $\mathcal{H}$ is public-coin[2] *if $\mathcal{H}.\mathsf{Gen}$ outputs a uniformly random string $k \leftarrow \{0,1\}^{s(\lambda)}$.*

Definition 6.4 (Correlation Intractability)**.** *For a given relation ensemble $R = \{R_\lambda \subseteq \{0,1\}^{n(\lambda)} \times \{0,1\}^{m(\lambda)}\}$, a hash family $\mathcal{H} = \{h_\lambda : \{0,1\}^{s(\lambda)} \times \{0,1\}^{n(\lambda)} \to \{0,1\}^{m(\lambda)}\}$ is said to be R-*correlation intractable with security *(s, δ) if for every s-size $\mathcal{A} = \{\mathcal{A}_\lambda\}$,*

$$\Pr_{\substack{k\leftarrow\mathcal{H}.\mathsf{Gen}(1^\lambda)\\x\leftarrow\mathcal{A}(k)}}\Big[\big(x, h(k, x)\big) \in R\Big] = O(\delta(\lambda)).$$

*We say that $\mathcal{H}$ is R-*correlation intractable with security *δ if it is (λ^c, δ)-correlation intractable for all $c > 1$. Finally, we say that $\mathcal{H}$ is R-*correlation intractable *if it is $(\lambda^c, \frac{1}{\lambda^c})$-correlation intractable for all $c > 1$.*

[2] Sometimes "public-coin" hash families are defined to be hash families whose security properties hold even when the adversary is given the random coins used to sample $k \leftarrow \mathcal{H}.\mathsf{Gen}(1^\lambda)$. For our purposes (e.g. ignoring compactness), this definition is equivalent to ours.

If $\mathcal{R}$ is a collection of relation ensembles, then $\mathcal{H}$ is said to be uniformly $\mathcal{R}$-correlation intractable *if for every polynomial-size $\mathcal{A}$, there exists a function $\nu(\lambda) = \text{negl}(\lambda)$ such that for every $R \in \mathcal{R}$,*

$$\Pr_{\substack{k \leftarrow \mathcal{H}.\mathsf{Gen}(1^\lambda) \\ x \leftarrow \mathcal{A}(k)}} \Big[(x, h(k,x)) \in R \Big] \leq \nu(\lambda).$$

6.3.1 Efficiently Searchable Relations

As in [CCH+19, PS19] we make use of hash functions that are correlation intractable for relations R with a *unique* output $y = f(x)$ associated to each input x, and such that $y = f(x)$ is an efficiently computable function of x.

Definition 6.5 (Unique Output Relation)**.** *We say that a relation R is a* unique output relation *if for every input x, there exists at most one output y such that $(x, y) \in R$.*

Definition 6.6 (Efficiently Searchable Relation, [CLW18])**.** *We say that a (necessarily unique-output) relation ensemble R is* searchable in (non-uniform) time T *if there exists a function $f = f_R : \{0,1\}^* \to \{0,1\}^*$ computable in (non-uniform) time T such that for any input x, if $(x, y) \in R$ then $y = f(x)$; that is, $f(x)$ is the unique y such that $(x, y) \in R$, provided that such a y exists. We say that R is* efficiently searchable *if it is searchable in time* $\mathsf{poly}(n)$*.*

In this work, we make use of the hash functions of [PS19], which are correlation-intractable for efficiently searchable relations under the LWE assumption (with polynomial modulus). Moreover, we use the fact that under subexponential LWE, the [PS19] hash family is in fact 2^{-m^δ}-correlation intractable for some $\delta > 0$.

Theorem 6.7 ([PS19])**.** *Assume the subexponential hardness of LWE. Then, there exists some $\delta > 0$ such that for all polynomial functions $(n(\cdot), m(\cdot), T(\cdot))$, there is a hash family $\mathcal{H} = \{h_\lambda : \{0,1\}^s \times \{0,1\}^n \to \{0,1\}^m\}$ that is $2^{-m(\lambda)^\delta}$-correlation intractable for all relations searchable in time T.*

6.4 Reverse Randomization-Compatible Trapdoor Σ-Protocols

In this section, we present a variant of "trapdoor Σ-protocols" [CCH$^+$19] that suffice for our transformation. The key differences as compared to the trapdoor Σ-protocols of [CCH$^+$19] are as follows.

- We require that the honestly generated CRS is uniformly random and that the "fake CRS" distribution is statistically close to uniform.
- We require malicious-verifier witness indistinguishability rather than just honest-verifier zero knowledge (these two properties are equivalent for protocols with polynomial-size challenge spaces and their parallel repetitions).

As we will explain, this can be achieved by instantiating the generic commitment scheme used in the [Blu86, FLS90] Σ-protocols using dual Regev encryption.

Definition 6.8 (Reverse Randomization-Compatible Trapdoor Σ-Protocol)**.** *We say that a* 3*-message protocol* $\Pi = (\mathsf{Gen}, P, V)$ *in the CRS model is a* reverse randomization-compatible trapdoor Σ-protocol *if there are p.p.t. algorithms* $\mathsf{TrapGen}, \mathsf{BadChallenge}$ *with the following syntax.*

- $\mathsf{TrapGen}(1^\lambda)$ *takes as input the security parameter. It outputs a common reference string* $\mathsf{crs} \in \{0,1\}^\ell$ *along with a trapdoor* td*.*
- $\mathsf{BadChallenge}(\mathsf{td}, \mathsf{crs}, x, \mathbf{a})$ *takes as input a trapdoor* td*, common reference string* crs*, instance* x*, and first message* $\mathbf{a}$*. It outputs a challenge* $\mathbf{e}$*.*

We additionally require the following properties.

- ***Witness Indistinguishability with Uniform CRS****.*
- ***CRS Indistinguishability:*** *The* crs *distribution output by* $\mathsf{TrapGen}(1^\lambda)$ *is statistically indistinguishable from the uniform distribution* U_ℓ*.*

- ***Efficient Special Soundness:*** *for every instance $x \notin L$ and for all $(\mathsf{crs}, \mathsf{td}) \leftarrow \mathsf{TrapGen}(1^\lambda)$, if $(\mathsf{crs}, x, \mathbf{a}, \mathbf{e}, \mathbf{z})$ is a valid transcript for Π, then $\mathbf{e} = \mathsf{BadChallenge}(\mathsf{td}, \mathsf{crs}, x, \mathbf{a})$.*

Remark 6.9. *Assuming the (polynomial) hardness of LWE, there is a reverse randomization-compatible trapdoor Σ-protocol for all of* **NP***.*

Proof. We instantiate Blum's Hamiltonicity protocol [Blu86] (or the [FLS90] Hamiltonicity protocol) in the CRS model using dual Regev encryption [GPV08]. The fact that these schemes satisfy efficient special soundness was already argued in [CCH+19]. Since dual Regev public keys are statistically indistinguishable from uniformly random, we are done. □

6.5 Constructing 2-Message WI

In this section, we show that correlation intractable hash functions for efficiently searchable relations (Section 6.3) can be combined with reverse randomization-compatible trapdoor Σ-protocols (Section 6.4) to obtain 2-message publicly verifiable WI arguments.

As we described in the introduction, this can be seen as an extension of the Dwork-Naor "reverse randomization" paradigm to the setting of comptuational soundness.

Construction 6.10 (2-Message WI Protocol)**.** *Let Π be a reverse randomization-compatible trapdoor Σ-protocol with the following three efficiency properties:*

- *Common reference strings have length $\ell(\lambda)$.*
- *Challenges have length $m(\lambda)$ for some polynomial function $m(\cdot)$.*
- *The algorithm $\mathsf{BadChallenge}(\tau, \mathsf{crs}, x, \mathbf{a})$ is computable by a size T circuit for some polynomial function $T(\lambda, n(\lambda))$.*

Moreover, let $\mathcal{H}$ denote a hash family that is $2^{-\ell}\mathrm{negl}(\lambda)$-correlation intractable for relations searchable in time T. We then define the following 2-message protocol $\widetilde{\Pi}$, which is a combination of the Fiat-Shamir transform (using $\mathcal{H}$) and [DN00]-style "reverse randomization."

- *Verifier message: the verifier samples λ common random strings $\mathsf{crs}_1, \ldots, \mathsf{crs}_t \xleftarrow{\$} \{0,1\}^\ell$ (for $t = 2\ell$) along with a hash key $k \leftarrow \mathcal{H}.\mathsf{Gen}(1^\lambda)$.*

- *Prover message: given an instance x, witness w, and verifier message $(\mathsf{crs}_1, \ldots, \mathsf{crs}_t, k)$, the prover does the following.*

  - *Sample a random string $\mathsf{crs}_P \xleftarrow{\$} \{0,1\}^\ell$ and set $\widetilde{\mathsf{crs}}_i = \mathsf{crs}_P \oplus \mathsf{crs}_i$.*

  - *For $1 \le i \le t$, compute $\mathbf{a}_i \leftarrow \Pi.P(\widetilde{\mathsf{crs}}_i, x, w)$, $\mathbf{e}_i = h(k, x||\mathbf{a}_i)$, $\mathbf{z} = \Pi.P(\widetilde{\mathsf{crs}}_i, x, w, \mathbf{a}_i, \mathbf{e}_i)$.*

  - *Output $(\mathbf{a}_i, \mathbf{e}_i, \mathbf{z}_i)_{i=1}^t$.*

- *The verifier accepts a transcript $\Big((\mathsf{crs}_i)_{i\le t}, k, x, \mathsf{crs}_P, (\mathbf{a}_i, \mathbf{e}_i, \mathbf{z}_i)_{i\le t}\Big)$ if for all i, $\mathbf{e}_i = h(k, x||\mathbf{a}_i)$ and $\Pi.V(\widetilde{\mathsf{crs}}_i, x, \mathbf{a}_i, \mathbf{e}_i, \mathbf{z}_i) = 1$.*

We claim that this construction yields a 2-message (publicly verifiable) WI argument system for **NP**. Completeness and public verifiability are clear by construction, so we proceed to prove that this protocol is both WI and sound.

Lemma 6.11. *Assuming that Π is WI, $\widetilde{\Pi}$ is also WI.*

Proof. This is identical to the [DN00] proof of witness indistinguishability, which we sketch here. Fix a malicious verifier V^* along with a statement, pair of witnesses, and auxiliary information (x, w_1, w_2, z). Then, consider the following views $\mathsf{view}^{(j)}$ for $0 \le j \le t$: for every j, let

$$\tau^{(j)} = \Big((\mathsf{crs}_i)_{i\le t}, k, x, \mathsf{crs}_P, (\mathbf{a}_i, \mathbf{e}_i, \mathbf{z}_i)_{i\le t}\Big)$$

and view$^{(j)} = (\tau^{(j)}, r)$, where:

- r is the internal randomness of V^*, and $\Big((\mathsf{crs}_i)_{i\le t}, k\Big) = V^*(x, w_1, w_2, z; r)$.

- For every i, $(\mathbf{a}_i, \mathbf{e}_i, \mathbf{z}_i)$ is computed using $\widetilde{\mathsf{crs}}_i := \mathsf{crs}_i \oplus \mathsf{crs}_P$. Moreover, it is computed using witness w_1 if and only if $j \ge i$ (and witness w_2 otherwise).

By construction, $\mathrm{view}^{(0)}$ is the view of V^* in an interaction with an honest prover using w_1, and $\mathrm{view}^{(t)}$ is the interaction between V^* and an honest prover using w_2. The computational indistinguishability of $\mathrm{view}^{(j)}$ and $\mathrm{view}^{(j+1)}$ for every j follows from the (malicious verifier) witness indistinguishability of Π. □

Lemma 6.12. *Assuming that $\mathcal{H}$ is $2^{-\ell}\mathrm{negl}(\lambda)$-correlation intractable for all relations searchable in time $T(\lambda, n(\lambda))$, $\widetilde{\Pi}$ is adaptively sound.*

Proof. Suppose that P^* is an efficient cheating prover that breaks the adaptive soundness of $\widetilde{\Pi}$ with non-negligible probability, meaning that

$$\Pr_{\substack{(\mathsf{crs}_1,\ldots,\mathsf{crs}_t),k \\ (x,\mathsf{crs}_P,\tilde{\pi})\leftarrow P(\mathsf{crs}_1,\ldots,\mathsf{crs}_t,k)}}[x \notin L \wedge V \text{ accepts } (x, \mathsf{crs}_1, \ldots, \mathsf{crs}_t, k, \mathsf{crs}_P, \tilde{\pi})] = \epsilon(\lambda)$$

for some non-negligible function $\epsilon(\cdot)$. We proceed to define a sequence of hybrid experiments where we change the underlying distributions and win conditions. Let $\mathsf{crs}^* \leftarrow \{0,1\}^\ell$ denote a uniformly random string of length ℓ sampled independently of the above random variables. Then, we have that

$$\Pr_{\substack{\mathsf{crs}^*,(\mathsf{crs}_1,\ldots,\mathsf{crs}_t),k \\ (x,\mathsf{crs}_P,\tilde{\pi})\leftarrow P(\mathsf{crs}_1,\ldots,\mathsf{crs}_t,k)}}[x \notin L \wedge V \text{ accepts } (x, \mathsf{crs}_1, \ldots, \mathsf{crs}_t, k, \mathsf{crs}_P, \tilde{\pi}) \wedge \mathsf{crs}_P = \mathsf{crs}^*] = \epsilon(\lambda)2^{-\ell}.$$

Next, in order to invoke correlation intractability, we need to argue that P^* must win while some $\widetilde{\mathsf{crs}}_i$ has a valid trapdoor. In order to have a uniform security reduction, we argue as follows. Since the CRS distribution output by $\mathsf{TrapGen}(1^\lambda)$ is statistically close to uniform, we know that there exists a set $\mathcal{S} \subseteq \{0,1\}^\ell$ of size $\frac{1}{2}2^\ell$ such that for every $\mathsf{crs} \in \mathcal{S}$, $\mathsf{TrapGen}(1^\lambda)$ outputs crs with probability at least $\frac{1}{2}2^{-\ell}$. By independence, we conclude that for every fixed string crs^*,

$$\Pr_{\mathsf{crs}_1,\ldots,\mathsf{crs}_t}[\mathsf{crs}^* \oplus \mathsf{crs}_i \notin \mathcal{S} \text{ for all i}] = 2^{-t} = 2^{-2\ell},$$

so we have that

$$\Pr_{\substack{\mathsf{crs}^*,(\mathsf{crs}_1,\ldots,\mathsf{crs}_t),k\\(x,\mathsf{crs}_P,\tilde{\pi})\leftarrow P(\mathsf{crs}_1,\ldots,\mathsf{crs}_t,k)}}[x \notin L \wedge V \text{ accepts } \wedge \mathsf{crs}_P = \mathsf{crs}^* \wedge \widetilde{\mathsf{crs}}_i \in \mathcal{S} \text{ for some } i] \geq \epsilon 2^{-\ell} - 2^{-2\ell}.$$

Picking a uniformly random $i^* \xleftarrow{\$} [t]$, we further see that

$$\Pr_{\substack{i^*,\mathsf{crs}^*,(\mathsf{crs}_1,\ldots,\mathsf{crs}_t),k\\(x,\mathsf{crs}_P,\tilde{\pi})\leftarrow P(\mathsf{crs}_1,\ldots,\mathsf{crs}_t,k)}}[x \notin L \wedge V \text{ accepts } \wedge \mathsf{crs}_P = \mathsf{crs}^* \wedge \widetilde{\mathsf{crs}}_{i^*} \in \mathcal{S}] \geq \frac{1}{4\ell}\epsilon 2^{-\ell}.$$

We next consider an alternate experiment in which the uniformly random crs_{i^*} is replaced by the string $\mathsf{crs}^* \oplus \overline{\mathsf{crs}}_{i^*}$ for $(\overline{\mathsf{crs}}_{i^*}, \mathsf{td}_{i^*}) \leftarrow \mathsf{TrapGen}(1^\lambda)$. Since every string in $\mathcal{S}$ has weight at least $\frac{1}{2}2^{-\ell}$ in the $\mathsf{TrapGen}$ crs distribution, we see that

$$\Pr_{\substack{i^*,\mathsf{crs}^*,\overline{\mathsf{crs}}_{i^*},(\mathsf{crs}_1,\ldots,\mathsf{crs}_t),k\\(x,\mathsf{crs}_P,\tilde{\pi})\leftarrow P(\mathsf{crs}_1,\ldots,\mathsf{crs}^*\oplus\overline{\mathsf{crs}}_{i^*},\ldots,\mathsf{crs}_t,k)}}[x \notin L \wedge V \text{ accepts } \wedge \mathsf{crs}_P = \mathsf{crs}^* \wedge \widetilde{\mathsf{crs}}_{i^*} \in \mathcal{S}] \geq \frac{1}{8\ell}\epsilon 2^{-\ell}.$$

Finally, we claim that this violates the $2^{-\ell}\mathrm{negl}(\lambda)$-correlation intractability of $\mathcal{H}$. Formally, an adversary $\mathcal{A}'$ can sample $i^*, (\overline{\mathsf{crs}}_{i^*}, \mathsf{td}_{i^*})$ and declare the relation

$$R_{\overline{\mathsf{crs}}_{i^*},\mathsf{td}_{i^*}} = \{(x||\mathbf{a}, \mathbf{e}) : \mathbf{e} = \mathsf{BadChallenge}(\mathsf{td}_{i^*}, \overline{\mathsf{crs}}_{i^*}, x, \mathbf{a}).\}$$

Then, upon receiving a hash key k, $\mathcal{A}'$ can sample crs^* and $(\mathsf{crs}_1, \ldots, \mathsf{crs}_t)$ itself and call $(x, \mathsf{crs}_P, \tilde{\pi}) \leftarrow P^*(\mathsf{crs}_1, \ldots, \mathsf{crs}^* \oplus \overline{\mathsf{crs}}_i, \ldots, \mathsf{crs}_t)$. Finally, $\mathcal{A}'$ outputs the pair $(x, \mathbf{a}_{i^*})$. Whenever $x \notin L$, $\mathsf{crs}_P = \mathsf{crs}^*$, and V accepts the output of P^* in the above experiment, by the efficient special soundness of Π, we will have that $(x, \mathbf{a}_{i^*}) \in R_{\overline{\mathsf{crs}}_{i^*},\mathsf{td}_{i^*}}$, completing the reduction. □

6.5.1 Parameter Settings and Instantiation

Combining Section 6.5 with Theorem 6.7 and Remark 6.9, we obtain the following LWE-based instantiation of 2-message publicly verifiable WI. Assume that LWE is $2^{-\lambda^\delta} \cdot \mathrm{negl}(\lambda)$-hard for some fixed $\delta > 0$.

- Using dual Regev encryption and the [Blu86] proof system for Hamiltonicity (repeated $\lambda^{\frac{2}{\delta}}$ times in parallel), there is a reverse randomization-compatible trapdoor Σ-protocol Π with a crs of size λ and challenges of length $\lambda^{\frac{2}{\delta}}$.
- Using Theorem 6.7, there is a hash family that is $2^{-\lambda^2} \cdot \mathrm{negl}(\lambda)$-correlation intractable for all relations that are searchable in time $T(\lambda)$ sufficient to compute the $\mathsf{BadChallenge}$ function associated to Π.
- Applying Section 6.5, we conclude that the protocol $\widetilde{\Pi}$ in Construction 6.10 (using these building blocks) is a 2-message publicly verifiaible WI argument system for **NP**. Moreover, it satisfies adaptive soundness (again by Section 6.5). Finally, since hash keys in the hash family $\mathcal{H}$ are pseudorandom, we conclude that another variant of $\widetilde{\Pi}$ (in which the verifier message is uniformly random) is a non-adaptively sound publicly-verifiable WI argument.

Part III

Multi-Input Correlation Intractability

Chapter 7

One-Way Product Functions and their Applications

7.1 Introduction

Cryptographically secure hash functions are a fundamental building block in cryptography. Some of their most ubiquitous applications include the construction of digital signature schemes [NY89], efficient CCA-secure encryption [BR93], succinct delegation of computation [Kil92], and removing interaction from protocols [FS87]. In their most general form, hash functions can be modeled as "random oracles" [BR93], in which case it is heuristically assumed that an explicitly described hash function H (possibly sampled at random from a family) behaves like a random function, as far as a computationally bounded adversary can tell.

One of the most basic properties one might desire from a hash function is *collision resistance*, which requires that a computationally bounded adversary, given an explicit (shrinking) function H, cannot find a pair of distinct inputs (x, y) such that $H(x) = H(y)$. Since their introduction [Dam88], collision-resistant hash functions have proved extremely useful in designing cryptographic primitives and protocols. As such, the following problem has received much attention in theoretical cryptography.

Question 7.1. *What are the assumptions from which collision-resistant hash func-*

tions can be built? In particular, can they be built from an arbitrary one-way function?

The question of building CRHFs from arbitrary one-way functions is particularly intriguing because OWFs are sufficient to construct a wide class of cryptographic primitives, including: pseudorandom generators [HILL99], pseudorandom functions [GGM84] and secret-key encryption, universal one-way hash functions [Rom90] and digital signatures, commitment schemes [Nao91], zero-knowledge proofs [GMW86], and garbled circuits [Yao86, LP09].

Unfortunately, all known constructions of CRHFs have required assumptions beyond general one-way functions, such as *structured* generic assumptions (e.g. the existence of claw-free pairs of permutations) or the hardness of specific problems (e.g. computing discrete logarithms or finding approximately short vectors on lattices). Even worse, there are strong negative results on the prospect of constructing CRHFs from arbitrary OWFs in the form of *black-box impossibility results.* The first such result is due to Simon [Sim98].

Theorem 7.2 ([Sim98], informal)**.** *There is an oracle relative to which no collision-resistant hash functions exist, but* exponentially secure *one-way permutations exist.*

In fact, CRHFs have proved to be an extremely frustrating primitive in theoretical cryptography, as they have evaded attempts to describe a hierarchy of cryptographic primitives (with "weaker" objects implied by the existence of "stronger" objects). In a stark demonstration of this problem, Asharov and Segev [AS15] proved that CRHFs are not even implied (in a black box[1] way) by one-way functions and the extremely powerful notion of indistinguishability obfuscation [BGI+01, GGH+13].

Theorem 7.3 ([AS15], informal)**.** *There is an oracle relative to which no collision-resistant hash functions exist, but* exponentially secure *one-way permutations and indistinguishability obfuscation exist.*

These negative results indicate substantial barriers to building CRHFs from OWFs

[1] "Black box" usage of IO and one-way functions is formalized through the notion of obfuscation for *oracle-aided* circuits. We refer the reader to [AS15] for details.

(or OWPs, or indeed from any of the vast array of primitives implied by IO and OWPs).

Collision resistance is also just *one* desirable property of random oracles, and our question above is a special case of the following more ambitious question.

Question 7.4. *Which random oracle properties can be guaranteed under standard cryptographic assumptions, and how weak can these assumptions be made?*

It is known that some random oracle properties are *not realizable* in the standard model [CGH98,GK03]. However, there has been a recent line of work [CCR16,KRR17, CCRR18] showing that under strong assumptions, many random oracle properties (specifically in the context of "single input correlation intractability") *can* be realized, and Question 7.4 in its full generality remains wide open.

7.1.1 Our Contributions

In this work, we make progress on all of the above questions by defining a natural strengthening of exponentially secure OWFs[2] that suffices for building CRHFs and more. An "uber" version of our assumption – which we state for the purpose of intuition but is quantitatively and qualitatively much stronger than what we actually require – states that for every $k = \mathsf{poly}(n)$, there exists an injective (polynomial-time computable) function $f : \{0,1\}^* \to \{0,1\}^*$ with the following "batch one-wayness" property: For every polynomial-size adversary $\mathcal{A}$, the probability that $\mathcal{A}(f(X_1), \ldots, f(X_k)) = (X_1, \ldots, X_k)$ for $X_1, \ldots, X_k \overset{\text{i.i.d.}}{\leftarrow} \{0,1\}^n$ is bounded by $2^{-kn} \cdot \mathsf{poly}(n)$.

Based on various significant *weakenings* of this uber-assumption, we construct:

- Collision-resistant hash families whose security against polynomial-time adversaries matches that of a random oracle.

- More generally, for every k, we construct hash families $\mathcal{H}$ that are "k-ary output intractable" (inspired by a related definition of Zhandry [Zha16]). Loosely

[2] Actually, OWFs where any *polynomial-time* algorithm can invert with only exponentially small probability

speaking, given $H \leftarrow \mathcal{H}$, it is computationally hard to find distinct inputs $X_1, \ldots, X_k$ such that $(H(X_1), \ldots, H(X_k))$ satisfy any fixed sparse relation R. The quantitative hardness that we achieve again matches that of a random oracle.

We are able to construct even stronger hash families if we additionally assume sub-exponentially secure indistinguishability obfuscation. This construction allows for applications including an instantiation of the Fiat-Shamir heuristic [FS87] for a natural class of interactive proofs.

Our main results and contributions are, in more detail, as follows.

Defining OWPFs.

We introduce the notion of a family of one-way k-product functions (k-OWPFs), which is a family of k-tuples of functions $(f_1, \ldots, f_k)$ that are jointly "extremely one-way". Such a family is most interesting when the hardness of inversion exceeds that of any individual f_i. For simplicity, suppose that each f_i is injective. In this case, we consider the assumption that no polynomial-time algorithm can recover $X_1, \ldots, X_k \overset{\text{i.i.d.}}{\leftarrow} \{0,1\}^n$ given $(f_1(X_1), \ldots, f_k(X_k))$ with probability better than δ. Ideally, this could be true for δ as large as $2^{-(k-o(k))n}$. We call this a δ-hardness assumption of *batch inversion* for $(f_1, \ldots, f_k)$.

The existence of such a family would follow from the following two conditions:

- A $\delta^{1/k}$-secure injective one-way function f, and
- An optimal *parallel repetition theorem* for the hardness of f, i.e. one which states that if a function f is (s, δ)-hard to invert, then its k-wise repetition f^k is (s, δ^k)-hard to invert.

While such a dream parallel repetition property likely does not hold for *general* f [DJMW12], the counterexample presented therein does not preclude a similar result for a broad class of functions f.

In fact, the parallel repetition framework described above yields a special kind of OWPF family: one in which all k functions $f_1, \ldots, f_k$ are equal. We say that such

OWPF families are *symmetric*. Another special case of interest, which we call a *one-way power family*, is a OWPF family of the form $\mathcal{F}^k$, meaning that the k functions $f_1, \dots, f_k$ are sampled independently at random from a fixed family $\mathcal{F}$.

Our constructions (that do not require obfuscation) are based directly on symmetric injective OWPFs as a building block rather than general OWPFs. We augment these constructions by providing generic transformations between different notions of OWPFs, including constructions of (weaker) symmetric OWPFs from (stronger) general OWPFs, and constructions of *injective* k-OWPFs from arbitrary k-OWPFs (with some security loss).

One of our main contributions in this work is initiating the study of OWPFs and establishing their basic properties. We expect that OWPFs will prove useful in future work.

On Extreme Hardness Amplification

For all of our constructions without obfuscation, we actually rely on *symmetric* OWPF families. That is, we want a family $\mathcal{F} = \{\mathcal{F}_n\}$ such that if we sample $f \leftarrow \mathcal{F}_n$ and $x_1, \dots, x_k \leftarrow \{0,1\}^n$, it is δ^k-hard to simultaneously invert $f(x_1), \dots, f(x_k)$. Clearly a necessary condition for this is that $\mathcal{F}$ is a δ-secure one-way function family. But is this sufficient? The answer in general is no, as we discuss next.

First of all, this type of attempted hardness amplification fails for any family whose functions have short trapdoors that enable polynomial-time inversion. Given $f, f(x_1), \dots, f(x_k)$, an adversary can simply guess the trapdoor for f, succeed with some small probability *that does not depend on* k, and conditioned on guessing correctly can efficiently invert $f(x_1), \dots, f(x_k)$.

It is natural to next consider *functions* (or ensembles of functions $\{f_n : \{0,1\}^n \to \{0,1\}^*\}_n$ indexed only by input length) that are secure against non-uniform adversaries, and in particular do not have any trapdoors. However, [DJMW12] present an example of a single one-way function f for which it is as easy to invert $f(x_1), \dots, f(x_k)$ as it is to invert a single $f(x)$. Although their counterexample heavily relies on the fact that there are multiple permissible solutions to each instance x, there is also evidence that parallel repetition sometimes fails to increase the security of *injective* one-way functions [Wic18].

Despite the above negative results, we emphasize that symmetric OWPFs only require direct products to amplify hardness for *specific* functions, rather than broad classes of functions. Moreover, one-way product functions may exist even if parallel repetition does not amplify the hardness of *any* function f beyond negligible. In particular, $f_1, \ldots, f_k$ may not all be the same function, and may be sampled from a joint distribution on k-tuples of functions. These observations leave us with at least two promising avenues towards constructing OWPF candidates:

1. Given the contrived nature of known counterexamples to one-way function parallel repetition, any "natural" δ-secure injective OWF family also serves as a candidate one-way power family with security roughly δ^k.

2. It may be possible to "fortify" any one-way function family $\mathcal{F}$ into a related family $\mathcal{F}'$ whose security *does* amplify to an extreme degree, yielding symmetric OWPFs.

Finally, we mention a concrete candidate symmetric OWPF family based on the *multiple discrete logarithm problem.* That is, in some group $\mathbb{G}_n$ of order $|\mathbb{G}_n| \approx 2^n$, the problem is to simultaneously compute k discrete logarithms $X_1, \ldots, X_k \overset{\text{i.i.d.}}{\leftarrow} [2^n]$ given input $(g, g^{X_1}, \ldots, g^{X_k})$, where g is a generator for $\mathbb{G}_n$. In [CK18], evidence for the hardness of computing multiple discrete logarithms is given in the form of lower bounds in the generic group model [Sho97]. In particular, [CK18] show that (in our language) k-batch inversion is nearly 2^{-kn}-hard for polynomial-time generic-group algorithms.

Constructions from OWPFs

Our first application of OWPFs is a construction of a collision-resistant hash family from suitably secure symmetric 2-OWPFs. Informally, we prove

Theorem 7.5. *Suppose that there exist symmetric injective* 2*-OWPFs with security* $2^{-n-\omega(\log n)}$*. Then, there exists a collision-resistant hash family.*

This type of OWPF does not follow in a black-box way from even exponentially-hard one-way permutations; this is how we avoid the [Sim98, AS15] impossibility results.

Through one of our generic transformations of OWPFs, we also obtain a construction that does not assume injectivity:

Theorem 7.6. *Suppose that there exist symmetric* 2*-OWPFs with security* $2^{-(1.6+\epsilon)n}$*. Then, there exists a collision-resistant hash family.*

Optimality and Implications of Theorem 7.5.

While we have explained how our result is not captured by the [Sim98, AS15] framework, one could question the necessity of this new OWPF assumption. For example, [AS15] only rules out black-box constructions of CRHFs from $2^{-\epsilon n}$-secure IO and one-way permutations (for $\epsilon = \frac{1}{50}$ in particular), and [Sim98] proves a quantitatively similar impossibility. What about assuming only $2^{-n/2}$-secure OWPs, which are weaker and more standard than our symmetric OWPFs? As a complementary result, we show that these are insufficient – we strengthen the Asharov-Segev analysis to rule out black box constructions from IO and even 2^{-n}-secure one-way permutations.

Theorem 7.7 (Extension of [AS15] Theorem 1.1, informal)**.** *There is no black-box construction of CRHFs from sub-exponentially secure IO, sub-exponentially secure OWPs, and OWPs that ppt algorithms* $\mathcal{A}$ *can invert with probability at most* $\mathrm{size}(\mathcal{A})^c \cdot 2^{-n}$ *for some absolute constant* c*.*

Theorem 7.7 indicates a sharp limit on directly improving Theorem 7.5; in the latter, we show that injective 2-OWPFs that are $2^{-n} \cdot \mathrm{negl}(n)$-hard to invert suffice for constructing CRHFs from IO, while the former result says that improving the $2^{-n} \cdot \mathrm{negl}(n)$ to $\frac{2^{-n}}{\mathrm{negl}(n)}$ is impossible for black-box constructions. In particular, for black-box constructions, exponentially secure one-way permutations (in the usual sense) are insufficient.

Extension to Output Intractability

Theorem 7.5 can be substantially generalized beyond collision-resistance. In particular, given a $2k$-ary relation, we consider the problem of finding $X_1, \ldots, X_k$ such that $(X_1, \ldots, X_k, H(X_1), \ldots, H(X_k)) \in R$ for $H \leftarrow \mathcal{H}_n$. If this problem is hard, then $\mathcal{H}$ is said to be multi-input correlation intractable for R, a notion due to [CGH98]. Collision-resistance is the special case when $k = 2$ and

$$R = \{(x_1, x_2, y_1, y_2) : (x_1 \neq x_2) \wedge (y_1 = y_2)\}.$$

Random oracles are correlation intractable for any *sparse* relation R – that is, as long as for every $\mathbf{x} = (x_1, \ldots, x_k)$, $\Pr_{\mathbf{Y} \leftarrow (\{0,1\}^{n-1})^k}[(\mathbf{x}, \mathbf{Y}) \in R] \leq \mathrm{negl}(n)$. In many applications, this correlation-intractability is the crucial property of a random oracle, and a fundamental theoretical question is whether it can be achieved by *concrete* hash families.

Despite the initial negative result of [CGH98], which ruled out correlation intractability for arbitrary (e.g., unbounded-arity) relations, there has been substantial work on constructing hash families that are correlation intractable for "bounded" single-input/output relations [CCR16, KRR17, CCRR18] as well as hash families that are "output intractable" [Zha16], that is, correlation intractable with respect to relations of the form "$(x_i \neq x_j$ for all $i \neq j) \wedge R(y_1, \ldots, y_k) = 1$."[3]

Using suitably secure k-OWPFs, we construct hash families that are output intractable for *all* sparse output relations (with known bounded arity). The quantitative intractability that we prove depends on the sparsity of the relation, similarly to the situation for a true random oracle. Equivalently, we rely on weaker assumptions to show correlation-intractability of sparser relations.

A simplified version of our result is as follows.

Theorem 7.8 (informal)**.** *Suppose that there exists a family of symmetric injective k-OWPFs with security $(s + \mathsf{poly}(n), \delta)$, let $m = m(n)$ denote any output length, and let $p = p(n)$ denote any sparsity. Then, there exists a hash family $\mathcal{H} = \{\mathcal{H}_{n,m(n)}\}$ that*

[3] [Zha16] considers a slightly different notion of output intractability. We elaborate on this later.

is output intractable, with security $(s, \delta \cdot p \cdot 2^{kn})$*, with respect to all* k*-ary relations of sparsity* p*.*

In particular, if the k-OWPF family has optimal (2^{-kn}) security, then the hash family constructed in Theorem 7.8 has output intractability matching that of a random oracle.

As an interesting special case, we note that Theorem 7.8 gives a construction of k-multi-collision resistant hash functions (formally introduced in [KNY17] and further studied in [BDRV18, BKP18, KNY18]) from symmetric injective k-OWPFs with security $2^{-n-k\log(k)} \cdot \text{negl}(n)$, an assumption that (up to a lower order term in the exponent) becomes weaker as k increases from 2 to any $o(\frac{n}{\log(n)})$. As any multi-collision-resistant hash family implies the existence of constant round statistically hiding commitments [BDRV18,KNY18], this yields constant round statistically hiding commitments from $2^{-n} \cdot \text{negl}(n)$-secure (injective and symmetric) k-OWPFs for any $k = o(\frac{n}{\log(n)})$. Unlike the assumptions required for collision resistance, this assumption would follow from optimal parallel repetition for *any polynomially secure (injective) one-way function.*

Combining OWPFs with Indistinguishability Obfuscation

Our results above, Theorem 7.5 and Theorem 7.8, are constructions of cryptographic hash families from (symmetric) OWPFs alone, and hence (partially) address the question of what hash families can be constructed from assumptions in the realm of one-wayness.

We additionally consider which hash families can be constructed in the plain model under stronger assumptions. Namely, we combine OWPFs with the powerful notion of indistinguishability obfuscation [BGI$^+$01, GGH$^+$13]. This line of reasoning yields another construction of CRHFs, and more generally a construction of multi-input correlation intractable hash functions for a broader class of relations than achieved by Theorem 7.8. In our IO-based construction, we are able to handle relations R which depend on both the input variables $\mathbf{x}$ and the output variables $\mathbf{y}$, as long as the relation R is efficiently *locally* samplable. Informally, we need to be able to efficiently sample a random output $\mathbf{Y}$ such that $(\mathbf{x}, \mathbf{Y}) \in R$ such that each output Y_i is sampled only knowing the corresponding input x_i (with arbitrary preprocessed shared randomness "between the variables").

Moreover, our construction is extremely simple and confirms typical intuition about obfuscation: our hash family is an obfuscated (puncturable) PRF $\mathcal{O}(F_s(\cdot))$. We only require the existence of suitably secure OWPFs in the security proof; they are not needed in the construction. This result extends the framework of [CCR16,KRR17] on constructing strong hash functions from obfuscation (and additional assumptions).

Our main result utilizing obfuscation (Theorem 7.57) is stated and proved in Section 7.6.3. The result is proved by viewing OWPFs themselves as a (weak) form of obfuscation: an injective k-OWPF $(f_1, \cdots, f_k)$ allows us to obfuscate *multi-point functions*, i.e., programs of the form

$$P_{x_1,\ldots,x_k}(x) = \begin{cases} i & x = x_i \text{ for some } i \\ 0 & \text{otherwise.} \end{cases}$$

Since this construction is oblivious to whether or not the OWPF family $\mathcal{F}$ is symmetric, this yields a construction of correlation intractable hash families (and in particular, of CRHFs) relying on weaker OWPF assumptions, at the cost of additionally assuming IO. That is, the assumptions on asymmetric OWPFs required here are quantitatively (and even qualitatively) weaker than those required without obfuscation, as we avoid the cost of converting asymmetric OWPFs into symmetric OWPFs.

As an interesting special case, the notion of correlation intractability that we achieve in Theorem 7.57 is powerful enough to capture nontrivial cases of the Fiat-Shamir paradigm for converting (constant round, public-coin) interactive proof systems into non-interactive argument systems. One such formal result is stated in Theorem 7.62, but the main intuition is that we can instantiate the Fiat-Shamir transform for proof systems with the property that a malicious prover can efficiently determine which verifier messages he can cheat on. This intuition captures protocols that follow the "commit-challenge-response" framework using a generic commitment scheme (which is the case that Theorem 7.62 handles). This approach yields a construction of NIZK argument schemes (in the common reference string model) through the Fiat-Shamir transform whose security relies on IO and the existence of exponentially secure one-way functions – no OWPF assumptions are needed in this case.

7.1.2 Related Work

Multi-Instance Security. There are a few other cryptographic constructions in the literature that are secure assuming a strong form of hardness amplification for one-way functions, or more generally some notion of multi-instance security. Several notable examples, although not a comprehensive listing, are as follows.

- In the context of password-based cryptography, [BRT12] study the multi-instance security of encryption schemes and key derivation functions. Their work is motivated by the common practice of "salting", which is intended to insure that the running time required for an adversary to compromise k users scales linearly with k.
- In the context of chosen ciphertext security, [RS09] consider the problem of simultaneously inverting $(f(x_1), \ldots, f(x_k))$ where $(x_1, \ldots, x_k)$ are sampled from a joint distribution (rather than i.i.d.). In contrast to our work, they only ask that the inversion probability should be $\mathrm{negl}(\lambda)$; that is, they do not ask for hardness to amplify. They show that *trapdoor functions* satisfying certain security properties of this flavor suffice to construct CCA-secure public key encryption.
- Inspired by Merkle puzzles, [BGI08] construct a public-key encryption scheme that allows for adversaries that run in time at most quadratically larger than that of the honest parties. They prove the security of their scheme under the assumption that there is a injective one-way function f, a polynomial $k = k(n)$, a constant $0 < \delta < \frac{1}{2}$, and a (randomized) "multi-source hard-core predicate" H such that for random $x_1, \ldots, x_k \leftarrow \{0,1\}^n$,

every algorithm running in time $2^{(1-\delta)n}$ on input $\big(f(x_1), \ldots, f(x_k), r\big)$ successfully guesses $H(x_1, \ldots, x_k, r)$ with advantage at most $2^{-\omega(n)}$.

- In concurrent and independent work, Bitansky and Lin [BL18] introduce the notion of an *amplifiable one-way function.* Roughly speaking, a one-way function f is (sub-exponentially) amplifiable if for all $k = \mathsf{poly}(n)$ there exists a hard-core predicate hcb for f and an efficiently computable *combiner* C such that given $(y_1 = f(x_1), \ldots, y_k = f(x_k))$ it is $2^{-k^{\epsilon}}$-hard (for $2^{n^{\epsilon}}$-time algorithms) to predict the combined hard-core bit $C(\mathsf{hcb}(x_1), \ldots, \mathsf{hcb}(x_k))$. The work [BL18] shows that such a one-way function is useful in the construction of a one message non-malleable commitment scheme.

Extremely Lossy Functions. [Zha16] introduces the notion of an extremely lossy function (ELF). In [Zha16], ELFs are used as a central building block to construct several hash families with strong security properties. In particular, they can be used to construct hash functions satisfying a notion of output intractability that is incomparable to we achieve in Section 7.5. Informally, [Zha16] considers the more general setting of $k+1$-ary relations $R(y_1, \ldots, y_k, w)$ with the property that for random $(y_1, \ldots, y_k)$, it is computationally hard to find a witness w for which $R(y_1, \ldots, y_k, w) = 1$ (where our notion would correspond to the case that for random $(y_1, \ldots, y_k)$, *no such witness exists*), and constructs hash functions that are correlation intractable for such relations R that are efficiently decidable.

The only current construction of ELFs relies on an exponentially strong DDH assumption. An interesting open question is whether OWPFs imply the existence of ELFs, or even ordinary (i.e. moderately) lossy one-way functions.

CRHFs from Extremely Strong LPN. Two recent works [YZW$^+$17, BLVW18] give constructions of CRHFs from the Learning Parity with Noise (LPN) problem in parameter settings that resemble an exponential hardness assumption. We note that one of the same works [BLVW18] proves that these particular LPN assumptions imply hardness in the complexity class $\mathrm{BPP}^{\mathrm{SZK}}$, placing this construction on similar complexity-theoretic ground as prior constructions from discrete logarithm and SIS. The LPN-based CRHFs are also provably broken in quasi-polynomial time, while our CRHF is plausibly as collision-resistant as a random oracle.

Single-Input Correlation Intractability. Correlation intractability [CGH98] is a clean but powerful property of random oracles that has drawn considerable interest, particularly for its relevance to the Fiat-Shamir transform [FS87, BR93]. Circumventing the negative results of [CGH98, GK03, BDG+13], there has been a recent line of work [CCR16, KRR17, CCRR18] on constructing (single input) correlation intractable hash functions and instantiating the Fiat-Shamir heuristic in the standard model, under strong assumptions. We build on this line of work, particularly the work of [KRR17], to achieve results for special cases of *multi-input* correlation intractability under weaker or incomparable assumptions than are required in these previous works.

CRHFs from IO and SZK-hardness. [BDV17] constructs CRHFs from indistinguishability obfuscation and any average-case hard problem in the complexity class $\mathsf{SZK}^{0,1}$. We consider SZK-hardness to be a "structured assumption" which makes it different from (even very strong) assumptions on injective one-way functions; indeed, the same work proves an Asharov-Segev-like impossibility result for constructing (even worst-case) hard SZK instances from IO and OWPs. A fascinating open question is whether OWPFs (with or without IO) imply SZK-hardness of any form.

7.1.3 Technical Overview

We now outline some of our constructions in more detail. In order to clearly demonstrate the power of OWPFs and our techniques, we focus on the following two special cases: constructing CRHFs from symmetric 2-OWPFs, and constructing CRHFs from IO and (asymmetric) injective 2-OWPFs.

Construction of CHRFs

For simplicity, we first assume that we have an ensemble of one-way permutations $\{f_n : \{0,1\}^n \to \{0,1\}^n\}$, where for every constant $c > 0$, double inversion is $2^{-n} \cdot n^{-c}$ hard for size-n^c adversaries. In this case, we construct a particularly simple CRHF: to sample a collision-resistant $H : \{0,1\}^n \to \{0,1\}^{n-1}$, first sample $P : \{0,1\}^n \to \{0,1\}^{n-1}$ from a pairwise independent hash family $\mathcal{P}$[4] $H = P \circ f_n$. This and similar constructions have proved very useful in prior works [NY89, PW08, Zha16].

We now sketch the proof of security. Assume for contradiction that some poly-size algorithm $\mathcal{A}$ finds collisions in H with probability $\epsilon = \epsilon(n)$. We show how to use $\mathcal{A}$ to simultaneously find $X_1^* = f_n^{-1}(Y_1^*)$ and $X_2^* = f_n^{-1}(Y_2^*)$ with probability roughly $\epsilon \cdot 2^{-n}$, given uniformly random $Y_1^*, Y_2^* \overset{\text{i.i.d.}}{\leftarrow} \{0,1\}^n$. Specifically, we will invoke $\mathcal{A}$ not on a uniformly sampled $H = P \circ f_n$, but on a differently defined $H = P_{\text{plant}} \circ f_n$, where P_{plant} is sampled from $\mathcal{P}$ *conditioned on* $P_{\text{plant}}(Y_1^*) = P_{\text{plant}}(Y_2^*)$.

Intuitively, we now argue (by a purely statistical argument) that (X_1^*, X_2^*) looks sufficiently like a *uniformly random* collision of H that $\mathcal{A}$ must output that exact collision with probability roughly $\epsilon \cdot 2^{-n}$. To make this intuition rigorous, suppose first that we ignore Y_1^* and Y_2^*, and simply invoke $\mathcal{A}$ on a randomly sampled $H = P \circ f_n$. Then with probability ϵ, $\mathcal{A}$ will find a collision (X_1, X_2) in H. Conditioned on this event, (X_1, X_2) will be *equal* to (X_1^*, X_2^*) with probability 2^{-2n}, for a total probability of $\epsilon \cdot 2^{-2n}$ that both events occur. But (X_1^*, X_2^*) is a collision in H with probability only $2^{-(n-1)}$. Thus, conditioning on this event (i.e., sampling $H = P_{\text{plant}} \circ f_n$ instead of $H = P \circ f_n$) boosts the probability that $\mathcal{A}$ outputs (X_1^*, X_2^*) to $\epsilon \cdot 2^{-2n} \cdot 2^{n-1} = \epsilon \cdot 2^{-n-1}$.

Therefore, the CRHF we constructed satisfies the standard notion of security: every polynomial-size adversary finds collisions with probability that is negligible in n. From stronger hardness assumptions on $\{f_n\}$, i.e. that double-inversion is $\delta(n)$-hard for size-$s(n)$ adversaries, one obtains a correspondingly more secure CRHF.

[4] We also require that the hash family is *programmable* at any two points, meaning that it is possible to sample a uniformly random $p \leftarrow \mathcal{P}$ subject to the condition that $p(y_1) = z_1$ and $p(y_2) = z_2$. See Definition 7.14.

Beyond Permutations and Injective One-Way Functions The above argument actually does not rely in any way on f_n being a permutation. It is, however, important that f_n is injective, so that all collisions in $P \circ f_n$ are due to P, and thus in some sense are randomly distributed.

We also show that the injectivity requirement can be traded off against a stronger hardness assumption. In fact, if $\{f_n\}$ is extremely secure to begin with, we can construct a family of functions which is statistically injective, and still nearly as secure.

For simplicity, we illustrate this transformation for *one-way functions.* Suppose that $\{f_n\}$ is $\delta(n)$-hard to invert for polynomial-time adversaries (think of $\delta(n) = 2^{-(1-o(1))n}$, although such extreme parameters are not necessary). We first observe that $\{f_n\}$ cannot be "extremely" non-injective; if one independently samples $X_1 \leftarrow \{0,1\}^n$ and $X_2 \leftarrow \{0,1\}^n$, then the probability that $f_n(X_1) = f_n(X_2)$ must be at most δ (otherwise one could break the security of f_n by random guessing). This can be leveraged to obtain a fully injective function (with some small error probability), as follows.

Set n to be any function of n' (think of $n(n') = 3n'$). Then define the ensemble of function families $\mathcal{F} = \{\mathcal{F}_{n'}\}$ as follows. To sample a function $f \leftarrow \mathcal{F}_{n'}$, sample $P : \{0,1\}^{n'} \to \{0,1\}^n$ from a pairwise independent hash family, and define $\tilde{f}_{n'} = f_n \circ P$. A simple pairwise independence argument shows that $\mathcal{F}$ is statistically injective, with failure probability at most $2^{2n'} \cdot \delta(n)$ (with the suggested parameters in mind, this is $2^{-(1-o(1))n'}$).

Security of $\mathcal{F}$ follows from observing that if an adversary cannot invert $f_n(X)$ with probability better than δ when sampling $X \leftarrow \{0,1\}^n$, then for any subset $\mathcal{X} \subseteq \{0,1\}^n$, the adversary cannot invert $f_n(X')$ with probability better than $\delta \cdot \frac{2^n}{|\mathcal{X}|}$ when sampling $X' \leftarrow \mathcal{X}$. With good probability ($1 - 2^{2n'-n}$, or with our suggested parameters $1 - 2^{-n'}$), it holds that $P : \{0,1\}^{n'} \to \{0,1\}^n$ is actually injective, so that inverting $f_n \circ P$ corresponds to inverting f_n when inputs are drawn from the uniform distribution on $\mathrm{Img}(P)$. The above discussion shows that this is $\delta \cdot 2^{n-n'}$-hard (or with our suggested parameters $2^{-(1-o(1))n'}$-hard) even for adversaries that are given arbitrary advice about P.

While the above description refers to the case of one-way functions (i.e. 1-OWPFs), similar arguments can be made for arbitrary OWPFs (with different quantitative tradeoffs), as discussed in Section 7.3.3.

Constructions Using Obfuscation

We now outline our general proof strategy – which we informally refer to as the *planting technique* – for all of our constructions based on IO, using collision resistance as an example. The planting technique is inspired by the recent work of Kalai, Rothblum, and Rothblum [KRR17] on instantiating the Fiat-Shamir heuristic using obfuscation.

For simplicity, we focus on hash functions that shrink by a single bit. Our construction is then simply an obfuscation $H \overset{\mathsf{def}}{=} \mathcal{O}(F_S)$ of a puncturable pseudorandom function $F_S : \{0,1\}^n \to \{0,1\}^{n-1}$, where $\mathcal{O}$ is an indistinguishability obfuscator. Recall that we also assume the existence of an injective but *not necessarily symmetric* 2-OWPF that cannot be inverted in polynomial time with probability better than $2^{-n-\omega(\log n)}$.

The proof of security then proceeds as follows. Assume for contradiction that some ppt algorithm $\mathcal{A}$ finds a collision (X_1, X_2) of H with non-negligible[5] probability ϵ. We then consider the behavior of $\mathcal{A}$ on an obfuscation of a *different* program H_{plant} which overrides the functionality of F_S with a hard-coded planted collision $H_{\text{plant}}(X_1^*) = H_{\text{plant}}(X_2^*) = Y^*$, for independent and uniformly random X_1^*, X_2^*, and Y^*. That is, the functionality of H_{plant} is

$$H_{\text{plant}}(x) \overset{\mathsf{def}}{=} \begin{cases} Y^* & \text{if } x = X_1^* \text{ or } x = X_2^* \\ F_S(x) & \text{otherwise.} \end{cases}$$

We then prove two contradictory claims.

[5] In fact, our approach readily generalizes to obtain exponentially-secure CRHFs, at the cost of quantitatively stronger computational assumptions.

Claim 1 (informal): The probability that $\mathcal{A}$ outputs (X_1^*, X_2^*) is approximately $\epsilon \cdot 2^{-n-1}$, i.e. $2^{-n-O(\log n)}$.

This claim is argued as follows.

(a) If $\mathcal{A}$ is given an obfuscation of a program H_{punc} that (in contrast to H_{plant}) overrides F_S with hard-coded mappings $X_1^* \mapsto Y_1^*$ and $X_2^* \mapsto Y_2^*$ for *independent* uniform $Y_1^*, Y_2^* \leftarrow \{0,1\}^{n-1}$, then the probability that $\mathcal{A}$ successfully produces a collision *and that collision is* (X_1^*, X_2^*) is very nearly $\epsilon \cdot 2^{-2n}$ by the security of $\mathcal{O}$ and F_S.

(b) (X_1^*, X_2^*) is only a valid collision of H_{punc} when $Y_1^* = Y_2^*$, so the probability that $\mathcal{A}$ outputs (X_1^*, X_2^*) conditioned on $Y_1^* = Y_2^*$ is approximately $\epsilon \cdot 2^{-2n} \cdot 2^{n-1} = \epsilon \cdot 2^{-n-1}$. But the distribution of H_{punc} conditioned on $Y_1^* = Y_2^*$ is exactly the distribution of H_{plant}.

Claim 2 (informal): The probability that $\mathcal{A}$ outputs (X_1^*, X_2^*) is $2^{-n-\omega(\log n)}$.

Since IO is the "best-possible" obfuscation [GR07], it suffices for there to exist *some* obfuscation of H_{plant} that hides (X_1^*, X_2^*). This would follow from a "special-purpose" obfuscator $\mathcal{O}'$ for membership testing in two-element sets (in our case $\{X_1^*, X_2^*\}$). The security property we need is that every ppt algorithm recovers (X_1^*, X_2^*) from $\mathcal{O}'(\{X_1^*, X_2^*\})$ with probability bounded by $2^{-n-\omega(\log n)}$. This is a variant of "point function obfuscation", a notion which was studied by [Can97, CMR98, Wee05]. Our variant (with uniformly random X_1^*, X_2^*) admits a particularly easy construction from injective 2-OWPFs – the obfuscation is $(W_1^* = f_1(X_1^*), W_2^* = f_2(X_2^*))$, and is evaluated on an input x as

$$\begin{cases} 1 & \text{if } f_1(x) = W_1^* \text{ or } f_2(x) = W_2^* \\ 0 & \text{otherwise.} \end{cases}$$

There are conceivably other ways to obtain this point function obfuscation, but for this particular construction, security is equivalent to the hardness of batch inverting (f_1, f_2).

7.1.4 Conclusions and Questions

In this work, we have introduced a new family of computational assumptions – namely, the existence of various flavors of one-way product functions (OWPFs). We find these assumptions to be clean, plausible, and useful.

In terms of power, OWPFs allow the construction of hash families that achieve several elusive random oracle-like properties. In particular, our black-box construction of CRHFs shows that OWPFs are more powerful than *black box usage* of exponentially-secure one-way functions.

OWPFs are also extremely plausible. Depending on s, δ, and k, we view (s, δ)-secure k-OWPFs as somewhere between standard and exponentially-secure one-way functions. The plausibility is supported by a concrete candidate instantiation – the discrete log problem, which is provably a nearly optimal OWPF in the generic group model.

Indeed, this particular combination of plausibility and usefulness gives us some hope that CRHFs can be constructed solely based on exponentially strong one-way functions. More generally, our results suggest a possible blueprint for circumventing black-box impossibility results from OWFs:

1. Build OWPFs from OWFs (using necessarily non-black-box techniques).

2. Build primitives in a black-box way from OWPFs.

One bonus of this approach is that it could result in constructions that are non-black-box only *in the security proof*, and thus has the potential for practical efficiency.

Independently, OWPFs satisfy several desirable properties for a cryptographic assumption. For example, for any family $\mathcal{F}$, the assumption "$\mathcal{F}$ is a k-OWPF" is a *search complexity assumption* [GK16]: for some efficiently sampleable distribution $\mathcal{D}$ and efficiently checkable relation $\mathcal{R}$, the assumption is equivalent to requiring that on input $x \sim \mathcal{D}$, every bounded-time algorithm has bounded probability of finding y such that $(x, y) \in \mathcal{R}$.

Questions

There remain many intriguing questions about the precise power of OWPFs. In particular:

- What are the complexity-theoretic implications of OWPFs? For example, do they imply hardness in SZK? We emphasize that all prior constructions of CRHFs have been from assumptions that imply (average-case) SZK hardness, but CRHFs themselves are not known to imply any sort of SZK hardness.

- What implies OWPFs? Is it possible to construct non-trivial k-OWPFs from previously studied cryptographic assumptions? Above we outlined an approach to *generically* constructing OWPFs, but it is also possible that OWPFs can be based on concrete, structured assumptions.

7.1.5 Organization

The rest of the paper is organized as follows. In Section 7.3, we define OWPFs and discuss the associated hardness assumptions, including a concrete candidate: the multiple discrete logarithm problem. We also prove generic reductions between OWPF notions. In Section 7.4, we present our construction of collision-resistant hash functions from (suitably secure) symmetric 2-OWPFs. In Section 7.5, we generalize the construction from Section 7.4 to obtain output intractable hash functions from symmetric OWPFs. In Section 7.6, we show that any (IO-)obfuscated puncturable PRF satisfies a broader notion of correlation intractability assuming that suitable OWPFs exist. This includes collision-resistant hash functions and output intractable hash functions from weaker OWPF assumptions as well as an instantiation of the Fiat-Shamir transform for "commit-challenge-response" proof systems. Finally, in Section 7.7, we formally state and prove Theorem 7.7, our complementary result showing that Theorem 7.5 is optimal.

7.2 Preliminaries

We write ppt to denote probabilistic polynomial-time. We say that two distribution ensembles $\{X_n\}$ and $\{Y_n\}$ are δ-indistinguishable if for all polynomial-sized circuit ensembles $\{\mathcal{A}_n\}$,

$$\left|\Pr\left[\mathcal{A}_n(X_n) = 1\right] - \Pr\left[\mathcal{A}_n(Y_n) = 1\right]\right| \leq O(\delta(n)).$$

For a relation R, we say that $R(x) = 1$ if $x \in R$ and $R(x) = 0$ otherwise.

For any primitive $\mathcal{P}$ whose security is parametrized by a pair $(s(\lambda), \delta(\lambda))$ (denoting time and advantage), we say that $\mathcal{P}$ is *polynomially secure* if $\mathcal{P}$ is $(\lambda^c, 1/\lambda^c)$-secure for all $c > 0$. We say that $\mathcal{P}$ is *sub-exponentially secure* if there exists some $\epsilon > 0$ such that $\mathcal{P}$ is $(2^{\lambda^\epsilon}, 2^{-\lambda^\epsilon})$-secure. We say that $\mathcal{P}$ is *δ-secure* if $\mathcal{P}$ is (λ^c, δ)-secure for all $c > 0$, and we say that $\mathcal{P}$ is *sub-exponential advantage-secure* if there exists some $\epsilon > 0$ such that $\mathcal{P}$ is 2^{-n^ϵ}-secure.

7.2.1 One-Way Functions

Definition 7.9 (One-Way Functions). *A polynomial-time computable function $f : \{0,1\}^* \to \{0,1\}^*$ is a (s, δ)-secure one-way function (OWF) if for every $\lambda \in \mathbb{N}$ and every circuit ensemble $\{\mathcal{A}_\lambda\}$ of size $|\mathcal{A}_\lambda| \leq s(\lambda)$, it holds that*

$$\Pr_{\substack{x \leftarrow \{0,1\}^\lambda \\ x' \leftarrow \mathcal{A}_\lambda(f(x))}} [f(x') = f(x)] \leq O(\delta(\lambda)).$$

Definition 7.10 (Families of One-Way Functions). *$\mathcal{F} = \{f_I : \mathcal{D}_I \to \mathcal{R}_I\}_{I \in \mathcal{I}}$ is a (s, δ)-secure family of one-way functions if there are ppt algorithms* $(\mathsf{Gen}, \mathsf{Samp})$ *and a deterministic polynomial-time algorithm* Eval *with the following syntax:*

- Gen *takes as input a security parameter 1^λ and outputs an index $I \in \mathcal{I}$.*

- Samp *takes as input an index $I \in \mathcal{I}$, and outputs $x \in \mathcal{D}_I$.*

- Eval *takes as input an index $I \in \mathcal{I}$ and $x \in \mathcal{D}_I$, and outputs $y = f_I(x)$.*

Additionally, there is a security requirement that for every circuit $\mathcal{A}$ of size $s(\lambda)$,

$$\Pr_{\substack{I \leftarrow \mathsf{Gen}(1^\lambda) \\ x \leftarrow \mathsf{Samp}(I) \\ x' \leftarrow \mathcal{A}(I, f_I(x))}} [f_I(x') = f_I(x)] \leq O(\delta(\lambda)).$$

For simplicity, we will only consider function families over the domain $\{0,1\}^\lambda$.

7.2.2 Cryptographic Hash Functions

The following definitions are adopted (with modification) from [Gol04].

Definition 7.11 (Cryptographic Hash Function)**.** *Fix a function $m : \mathbb{N} \to \mathbb{N}$ such that $1^{m(n)}$ is computable from 1^n in polynomial time. A family of functions*

$$\mathcal{H} = \{h_I : \{0,1\}^{n(I)} \to \{0,1\}^{m(n(I))}\}_{I \in \mathcal{I}}$$

is a (cryptographic) hash family *if there is a ppt algorithm* Gen *and a deterministic polynomial-time* Eval *such that:*

- *(Efficient Sampling) On input 1^n, Gen outputs an index $I \in \mathcal{I}$ such that $n(I) = n$.*
- *(Admissible Indexing – technical[6]) There is a polynomial-time algorithm that when given $I \leftarrow \mathsf{Gen}(1^n)$ as input, outputs 1^n.*
- *(Efficient Evaluation) For all $I \in \mathcal{I}$ and all $x \in \{0,1\}^{n(|I|)}$, $\mathsf{Eval}(I, x) = h_I(x)$.*

The above definition details the *functionality* of a hash function; there are several security notions that one could require. We first focus on the notion of *k-collision-resistance*, recovering the usual definition of a collision-resistant hash family when $k = 2$.

[6] Roughly, we would like the notion of polynomial-time in the *description length* of a hash function to coincide with the notion of polynomial-time in the security parameter

Definition 7.12 (k-collision-resistance)**.** *A family of cryptographic hash functions*

$$\mathcal{H} = \{h_I : \{0,1\}^{n(I)} \to \{0,1\}^{m(n(I))}\}_{I \in \mathcal{I}}$$

is a (length-restricted) k-collision-resistant hash family (k-CRHF) with security $\delta = \delta(m(\cdot))$ if the following two conditions hold.

- *(Shrinking) $m(n) \leq n - \log(k)$.*
- *(k-Collision-Resistance) For all polynomial-size circuits $\mathcal{A}$,*

$$\Pr_{\substack{I \leftarrow \mathsf{Gen}(1^n) \\ (X_1,\ldots,X_k) \leftarrow \mathcal{A}_n(I)}} [h_I(X_1) = \ldots = h_I(X_k) \text{ but } X_1, \ldots, X_k \text{ are all distinct}] \leq O(\delta(m(n))).$$

We say that $\mathcal{H}$ is polynomially secure *if $\mathcal{H}$ is $1/m(n)^c$-secure for all $c > 0$.*

Definition 7.13 (Universal One-Way Hash Families)**.** *A universal one-way hash family (UOWHF) is a family of cryptographic hash functions*

$$\mathcal{H} = \{h_I : \{0,1\}^{n(|I|)} \to \{0,1\}^{m(n(|I|))}\}_{I \in \mathcal{I}}$$

as in Definition 7.11 which are shrinking as in Definition 7.12, but (2-)collision-resistance is weakened to require only that for all polynomial-size circuits $\mathcal{A}_0, \mathcal{A}_1$, there is a negligible function $\nu(\cdot)$ such that

$$\Pr_{\substack{(X,\mathsf{st}) \leftarrow \mathcal{A}_0(1^n) \\ I \leftarrow \mathsf{Gen}(1^n) \\ X' \leftarrow \mathcal{A}_1(I,\mathsf{st})}} [h_I(X) = h_I(X') \wedge X \neq X'] \leq \nu(m(n)).$$

Finally, we define k-wise independent hash functions, which exist unconditionally.

Definition 7.14 ((Programmable) k-wise Independent Hash Functions)**.** *A family of k-wise independent hash functions is a family of hash functions*

$$\mathcal{H} = \{h_I : \{0,1\}^{n(|I|)} \to \{0,1\}^{m(n(|I|))}\}_{I \in \mathcal{I}}$$

as in Definition 7.11 with the property that for every collection $x_1, \ldots, x_k \in \{0,1\}^n$ of distinct inputs, and every collection $y_1, \ldots, y_k \in \{0,1\}^m$ of (not necessarily distinct) outputs, we have

$$\Pr_{I \leftarrow \mathsf{Gen}(1^n)}[h_I(x_i) = y_i \textit{ for all } i] = \frac{1}{2^{km}}.$$

Moreover, we say that $\mathcal{H}$ is programmable *if there is an efficient sampling algorithm* $\mathrm{CondGen}(\mathbf{x}, \mathbf{y})$ *with the property that for every $\mathbf{x} = (x_1, \ldots, x_k)$ and $\mathbf{y} = (y_1, \ldots, y_k)$ as above,* $\mathrm{CondGen}(\mathbf{x}, \mathbf{y})$ *samples from the distribution of $I \leftarrow \mathsf{Gen}(1^n)$ subject to the condition that $h_I(x_i) = y_i$ for all i.*

7.3 One-Way Product Functions: Definitions and Reductions

In this section, we define one-way product functions and their associated batch inversion problems, we discuss the discrete log problem as a concrete candidate, and we establish reductions between different notions of OWPFs.

Definition 7.15 (k-Batch Inversion, k-OWPFs)**.** *Let $\mathcal{F}$ be a family of k-tuples of functions, i.e.,*

$$\mathcal{F} = \{(f_{1,I}, f_{2,I}, \ldots, f_{k,I})\}_{I \in \mathcal{I}},$$

*where each $f_{i,I} : D_{i,I} \to R_{i,I}$. We say that k-*batch inversion *is* $(s(\lambda), \delta(\lambda))$-hard *for $\mathcal{F}$ (equivalently $\mathcal{F}$ is a* (s, δ)-secure k-OWPF family*) if for every size-$s(\lambda)$ circuit $\mathcal{A}$, we have*

$$\Pr\left[\forall i \in [k], f_{i,I}(X'_i) = f_{i,I}(X_i)\right] \leq O(\delta(\lambda))$$

in the probability space defined by sampling

1. $I \leftarrow \mathsf{Gen}(1^\lambda)$.
2. *For $i = 1, \ldots, k$, $X_i \leftarrow \mathsf{Samp}(I_i)$.*
3. $(X'_1, \ldots, X'_k) \leftarrow \mathcal{A}(I, f_{1,I}(X_1), \ldots, f_{k,I}(X_k))$.

In the special case $k = 2$, we refer to 2-batch inversion as "double inversion".

For the rest of this paper, we will work only over a fixed domain $\mathcal{D} = \{0,1\}^\lambda$ for simplicity.

Remark 7.16. *For any family $\mathcal{F}$ as above, if any of the families $\mathcal{F}_i := \{f_{i,I}\}_{I \in \mathcal{I}}$ is a family of (s, δ)-secure one-way functions, then k-batch inversion is (s, δ)-hard for $\mathcal{F}$. That is, (s, δ)-secure k-OWPFs follow from (s, δ)-secure OWFs.*

Given Remark 7.16 above, we note that batch inversion assumptions are most naturally suited to the setting where $\delta \le 2^{c\lambda}$ for some c, i.e., δ is *exponentially small*. Moreover, the batch inversion problem is quite plausibly $(\mathsf{poly}(\lambda), \delta)$-hard for $\delta < 2^{-\lambda}$, i.e. where δ is so small that any one-way function can trivially be inverted with probability δ (by outputting a uniformly random guess).

For any family of k-tuples of functions $\mathcal{F}$, we now state the strongest quantitative assumption that is plausible regarding batch inversion for $\mathcal{F}$ (and in particular, such families exist in the random oracle model).

Definition 7.17 (Optimal Batch Inversion Assumption for $\mathcal{F}$)**.** *There exists a universal constant c such that for every function $s = s(\lambda)$, the k-batch inversion problem for $\mathcal{F}$ is $(s(\lambda), s(\lambda)^{ck} 2^{-k\lambda})$-hard.*

This assumption, while not technically falsifiable in the framework of [Nao03, GW11], is still "morally" falsifiable, and in particular is a complexity assumption in the framework of [GK16].

We now consider two important special cases of k-OWPFs.

Definition 7.18 (Symmetric k-OWPFs)**.** *We say that a family $\mathcal{F}'$ of k-OWPFs is* symmetric *if for all indices $I \in \mathcal{I}$, we have $f_{1,I} = f_{2,I} = \ldots = f_{k,I}$. In other words, $\mathcal{F}'$ is a family of symmetric k-OWPFs if there is a family $\mathcal{F} = \{f_I\}_{I \in \mathcal{I}}$ such that (1) $\mathcal{F}' = \{(f_I, f_I, \ldots, f_I)\}_{I \in \mathcal{I}}$ and (2) $\mathcal{F}'$ is a family of k-OWPFs.*

As described in the introduction, the existence of a family of δ-secure symmetric k-OWPFs would follow from the following two conditions:

- A $\delta^{1/k}$-secure family $\mathcal{F}$ of injective one-way functions, and
- An optimal *parallel repetition theorem* for the hardness of $\mathcal{F}$, i.e. one which states that if a function $f \leftarrow \mathcal{F}$ is (s, δ)-hard to invert, then its k-wise repetition f^k is (s, δ^k)-hard to invert.

However, such a "dream parallel repetition theorem" (even for a specific family $\mathcal{F}$) is not required for δ-secure k-OWPFs to exist. As an example, for any $k \ll \frac{n}{\log(n)}$, consider the question of obtaining 2^{-n}-secure symmetric k-OWPFs; this is a parameter setting of interest for the application of k-multi-collision resistant hash functions. The existence of such a family would also follow from a 2^{-cn}-secure injective OWF family $\mathcal{F}$, along with a much weaker parallel repetition theorem for the hardness of $\mathcal{F}$; hardness would only have to amplify by a factor of $\frac{1}{c}$ in the exponent after k repetitions.

Definition 7.19 (One-Way Power Families). *We say that a function family $\mathcal{F}'$ is a* one-way power family *if there is a family $\mathcal{F} = \{f_I\}_{I \in \mathcal{I}}$ such that (1) $\mathcal{F}' = \mathcal{F}^k = \{(f_{I_1}, f_{I_2}, \ldots, f_{I_k})\}_{(I_1,\ldots,I_k) \in \mathcal{I}^k}$ and (2) $\mathcal{F}'$ is a family of k-OWPFs.*

In constrast to symmetric OWPFs, (s, δ)-secure one-way power families follow from the following two conditions.

- A $\delta^{\frac{1}{k}}$-secure family $\mathcal{F}$ of injective one-way functions, and
- A *different* form of (optimal) parallel repetition for $\mathcal{F}$, i.e. one which states that if a function $f \leftarrow \mathcal{F}$ is (s, δ)-hard to invert, then k independently sampled functions $f_1, \ldots, f_k \leftarrow \mathcal{F}$ are (s, δ^k) hard to simultaneously invert.

This alternative form of parallel repetition avoids the issue of breaking f^k by brute-forcing a short trapdoor for f; in the case of one-way power families, each of the k functions would have a different trapdoor.

We again emphasize that these optimal parallel repetition results are far stronger than what is required to obtain many of our applications of OWPFs.

7.3.1 Concrete Candidate: Discrete Logarithm

The optimal batch inversion assumption above, even in the setting of symmetric k-OWPFs, is supported by the work of [CK18], who consider the *multiple discrete logarithm problem*:

Definition 7.20 (Multiple Discrete Logarithm Problem, informal)**.** *Given a sequence of groups $\mathcal{G} = \{G_\lambda, \lambda \in \mathbb{N}\}$ (with efficiently computable operations and sampling algorithms), the multiple discrete logarithm problem is, given as input $(g, y_1, \ldots, y_k) = (g, g^{x_1}, \ldots, g^{x_k})$ (for uniformly random $x_1, \ldots, x_k$), to return all k discrete logarithms $(x_1, \ldots, x_k)$.*

In [CK18], evidence for the hardness of computing multiple discrete logarithms is given in the form of lower bounds in the generic group model [Sho97]. Specifically, they show

Theorem 7.21 ([CK18] Theorem 8, interpreted)**.** *Any generic group algorithm for the multiple discrete logarithm problem running in time T in a group of order $\Theta(2^\lambda)$ has success probability at most $T^{2k}2^{-\lambda k}\mathsf{poly}(\log(T), \lambda, k)^k$.*

In other words, the optimal batch inversion assumption holds for generic group discrete logarithms. Moreover, the best known algorithms for multiple discrete logarithm over elliptic curve groups are these generic algorithms, and hence the optimal batch inversion assumption over elliptic curve groups is plausible. This yields a candidate family of symmetric k-OWPFs satisfying optimal batch inversion hardness.

The multiple discrete logarithm problem (as defined above) provides a candidate *symmetric* OWPF family. We could alternatively consider the problem of computing k discrete logarithms, *each over an entirely different group*; this would constitue a candidate (asymmetric) OWPF family. In the special case where the k groups are sampled independently at random from some family, this would constitute a candidate one-way power family.

7.3.2 OWPFs that are Sufficient for CRHFs

In order to build collision-resistant hash functions, we do not need the optimal double inversion assumption, but the following weaker assumption (albeit for injective functions).

Conjecture 2. *There is a* $2^{-\lambda-\omega(\log \lambda)}$*-secure injective* 2*-OWPF family.*

That is, we require that double inversion is $2^{-\lambda} \cdot \mathrm{negl}(\lambda)$-hard (rather than $2^{-2\lambda}$-hard) for polynomial time algorithms. Our correlation intractability results are also achieved under assumptions significantly weaker than the optimal assumption (we state the necessary assumptions in Section 7.5 and Section 7.6.3).

In the rest of this section, we describe how to obtain OWPFs of a special form – either symmetric, injective, or both – from more general OWPFs through a few different transformations. We consider these transformations with the goal of obtaining important applications of (symmetric injective) OWPFs, such as multi-collision-resistant hash functions, in mind.

7.3.3 From OWPFs to Injective OWPFs

Our symmetric OWPF-based constructions most naturally work with (statistically) injective symmetric OWPFs, but an arbitrary OWPF family may be far from injective. To handle this issue, we present a modular transformation which converts, with some security loss, any symmetric OWPF family into a (statistically) injective symmetric OWPF family. In the rest of the paper, we will often assume that our symmetric OWPF families are statistically injective, which can be guaranteed using this transformation.

In addition, we provide a second transformation which converts arbitrary OWPF families into (statistially) injective OWPF families with the property that one-way power families (Definition 7.19) are mapped to one-way power families under this transformation. The security loss in the "one-way power family" case matches the security loss in the symmetric case, while the security loss for general OWPFs is quantitatively worse (for reasons that will become clear). This transformation allows for additional constructions from general OWPFs (and one-way power families), both with and without obfuscation.

We begin with the symmetric case. Let $\mathcal{F} = \{\{(f_I : \{0,1\}^\lambda \to \{0,1\}^*)^k\}_{I \in \mathcal{I}_\lambda}\}_{\lambda \in \mathbb{N}}$ be a family of symmetric OWPFs. We consider the following family $\mathcal{F}'$ of OWPFs with input domain $\{0,1\}^n$. We show that with an appropriate choice of n, it is a *statistically injective* k-OWPF.

Construction 7.22. *Given a family of OWPFs $\mathcal{F}$ and a function $\lambda = \mathsf{poly}(n)$, define the OWPF family $\mathcal{F}'$ as follows. Let $\mathcal{H}_n : \{0,1\}^n \to \{0,1\}^\lambda$ be a pairwise independent hash family.*

$\mathcal{F}'.\mathsf{Gen}$: *On input 1^n sample $H \leftarrow \mathcal{H}_n$, sample $I \leftarrow \mathcal{I}$, and output (H, I).*

$\mathcal{F}'.\mathsf{Samp}$: *On input (H, I), output a uniformly random $W \leftarrow \{0,1\}^n$.*

$\mathcal{F}'.\mathsf{Eval}$: *On input $\Big((H,I), W\Big)$, output $f_I(H(W))$.*

We use the notation $f'_{I,H}$ as shorthand for a member of the family $\mathcal{F}'$. We first describe the parameter settings in which $\mathcal{F}'$ is statistically injective. Let $\textsc{Inj}$ denote the event (over the randomness of $\mathcal{F}'.\mathsf{Gen}$) that the function $f'_{I,H}$ is injective.

Claim 7.22.1. *Suppose that $\mathcal{F}$ is a family of δ-secure k-OWPFs. Then, the probability of $\neg\textsc{Inj}$ is at most $2^{2n} \cdot \delta(\lambda)^{\frac{1}{k}}$.*

Proof. Let N denote the random variable equal to the number of distinct pairs (w_1, w_2) for which $f_I(H(w_1)) = f_I(H(w_2))$. Then we have $\Pr[\neg\textsc{Inj}] = \Pr[N \geq 1]$, which by Markov's inequality is at most $\mathbb{E}[N]$.

Let $C(w_1, w_2)$ denote the event that $f_I(H(w_1)) = f_I(H(w_2))$, and let $1_{C(w_1,w_2)}$ denote the corresponding indicator random variable, so that $N = \sum_{w_1 \neq w_2} 1_{C(w_1,w_2)}$. For every $w_1 \neq w_2$ and every i, the pairwise independence of $\mathcal{H}_n$ implies that

$$\mathbb{E}[1_{C(w_1,w_2)} | I = i] = \Pr_{x_1, x_2 \overset{\text{i.i.d.}}{\leftarrow} \{0,1\}^\lambda} \left[f_i(x_1) = f_i(x_2)\right].$$

We call the latter probability the collision probability of i, and denote it by $\mathsf{CP}(i)$. By the above, $\mathbb{E}[N | I = i] = \binom{2^n}{2} \cdot \mathsf{CP}(i)$.

In order for the trivial attack (guess $x_1, x_2, \ldots, x_k$ uniformly at random) to not violate the δ-security of $\mathcal{F}$ as a k-OWPF, it must be that

$$\mathbb{E}\left[\mathsf{CP}(I)^k\right] \leq \delta(\lambda). \tag{7.1}$$

Thus, we have

$$\begin{aligned}
\Pr[\neg\textsc{Inj}] &\leq \mathbb{E}[N] \\
&= \mathbb{E}\Big[\mathbb{E}[N|I]\Big] \\
&= \mathbb{E}\left[\binom{2^n}{2} \cdot \mathsf{CP}(I)\right] \\
&= \binom{2^n}{2} \cdot \mathbb{E}[\mathsf{CP}(I)] \\
&\leq 2^{2n} \cdot \delta(\lambda)^{\frac{1}{k}},
\end{aligned}$$

where the last inequality follows from Jensen's inequality together with Eq. (7.1). □

Having analyzed the injectivity of $\mathcal{F}'$, we now argue about its security.

Proposition 7.23. *If $\mathcal{F}$ is a family of $\big(s(\lambda) + \mathsf{poly}(\lambda), \delta(\lambda)\big)$-secure k-OWPFs, then for any non-constant $\lambda = \mathsf{poly}(n)$, it holds that $\mathcal{F}'$ is an $\big(s(\lambda), \delta'(n)\big)$-secure family of k-OWPFs, where $\delta'(n)$ is the maximum of:*

- $\delta(\lambda) \cdot \left(\frac{2^{-n}}{2^{-\lambda}}\right)^k$ *and*
- $2^{-\lambda} \cdot 2^{2n}$.

Given the bounds proved in Proposition 7.23 and Claim 7.22.1, we now consider the special case $\delta(\lambda) = 2^{-\theta k n}$ for intuition. In one reasonable setting of parameters, we can choose

$$\lambda(n) = \frac{k+2}{(1-\theta)k + \theta} n,$$

which yields a OWPF family with security and non-injectivity probability both bounded by

$$\delta'(n) = 2^{-\frac{\theta(k+2)}{(1-\theta)k+\theta}n}.$$

As an example, this yields a $2^{-n} \cdot \mathrm{negl}(n)$-secure injective symmetric k-OWPF (which is sufficient for k-multi collision-resistant hash functions when $k = o(\frac{n}{\log(n)})$ for any $\theta > \frac{3k}{4k-1}$. This implies a construction of collision-resistant hash functions from $2^{\frac{2n}{7}-2n}$-secure symmetric 2-OWPFs.[7]

Proof of Proposition 7.23. Let $P^{(n)}$ denote the distribution of $(H, \mathbf{X})$ in the experiment defined by independently sampling $H \leftarrow \mathcal{H}_n$ and $\mathbf{W} \leftarrow (\{0,1\}^n)^k$, and then defining $X_1 = H(W_1), \ldots, X_k = H(W_k)$. Specifically, we have

$$P^{(n)}(h, \mathbf{x}) = \Pr_{H \leftarrow \mathcal{H}_n}[H = h] \cdot \frac{\prod_{i=1}^{k} \Big|\{w : h(w) = x_i\}\Big|}{2^{kn}}. \tag{7.2}$$

Let $Q^{(n)}$ denote the distribution of $(H, \mathbf{X})$ in the experiment defined by independently sampling $H \leftarrow \mathcal{H}_n$ and $\mathbf{X} \leftarrow (\{0,1\}^\lambda)^k$. Specifically, we have

$$Q^{(n)}(h, \mathbf{x}) = \Pr_{H \leftarrow \mathcal{H}_n}[H = h] \cdot \Pr_{\mathbf{X} \leftarrow (\{0,1\}^\lambda)^k}[\mathbf{X} = \mathbf{x}]. \tag{7.3}$$

We first note that if h is an injective function, then $P^{(n)}(h, \mathbf{x}) \leq 2^{k\lambda} 2^{-kn} Q^{(n)}(h, \mathbf{x})$ for all $\mathbf{x} \in \{0,1\}^\lambda$.

Now, to prove Proposition 7.23, consider the event $\textsc{Win}_n$ that consists of the outcomes $(I, h, \mathbf{x})$ for which $\mathcal{A}_n\Big(I, h, f_I(x_1), \ldots, f_I(x_k)\Big)$ outputs $(w_1, \ldots, w_k)$ such that for each $i \in [k]$, $f_I\Big(h(w_i)\Big) = f_I(x_i)$. Now suppose that $\mathcal{A}$ wins the k-inversion game for $\mathcal{F}'$ with probability greater than $2\delta'$; this exactly means that $P(\textsc{Win}_n) \geq 2\delta'$. Then, consider the algorithm $\mathcal{B}_n$ that on input $Y_1, \ldots, Y_k$ samples $H \leftarrow \mathcal{H}_n$, computes $(W_1, \ldots, W_k) \leftarrow \mathcal{A}_n(H, Y_1, \ldots, Y_k)$, and outputs $\Big(H(W_1), \ldots, H(W_k)\Big)$. The probability of $\mathcal{B}_n$ winning the k-inversion game for f (on security parameter $\lambda(n)$) is

[7] For the specific application of polynomially secure (M)CRHFs, one can tweak parameters differently and obtain a construction from $2^{\frac{-2k}{3k-1}kn}$-secure symmetric k-OWPFs. This is because it suffices to have non-injectivity probability $\mathrm{negl}(n)$ for the later construction to work.

just $Q^{(n)}(\textsc{Win}_n)$. However, we now note that

$$\begin{aligned} Q^{(n)}(\textsc{Win}_n) &\geq Q^{(n)}(\textsc{Win}_n \wedge h \text{ injective}) \\ &\geq 2^{k(n-\lambda)} P^{(n)}(\textsc{Win}_n \wedge h \text{ injective}) \\ &\geq 2^{k(n-\lambda)} \left(P^{(n)}(\textsc{Win}_n) - \Pr_{H \leftarrow \mathcal{H}_n}[H \text{ not injective}] \right) \\ &\geq 2^{k(n-\lambda)}(2\delta' - 2^{-\lambda+2n}) \\ &\geq 2^{k(n-\lambda)}\delta' \geq \delta. \end{aligned}$$

This contradicts the security of $\mathcal{F}$, so we have proved Proposition 7.23. □

Having handled the symmetric case, we now turn to our second transformation.

Construction 7.24. *Given a family of OWPFs $\mathcal{F}$ and a function $\lambda = \mathsf{poly}(n)$, define the OWPF family $\mathcal{F}'$ as follows. Let $\mathcal{H}_n : \{0,1\}^n \to \{0,1\}^\lambda$ be a pairwise independent hash family.*

$\mathcal{F}'.\mathsf{Gen}$: *On input 1^n sample $H_1, \dots H_k \leftarrow \mathcal{H}_n$ independently at random, sample $I \leftarrow \mathcal{I}$, and output $(H_1, \dots, H_k, I)$.*

$\mathcal{F}'.\mathsf{Samp}$: *On input $(j, (H_1, \dots, H_k, I))$, output a uniformly random $W \leftarrow \{0,1\}^n$.*

$\mathcal{F}'.\mathsf{Eval}$: *On input $\Big(j, (H_1, \dots, H_k, I), W\Big)$, output $f_{j,I}(H_j(W))$.*

Note that if $\mathcal{F}$ is a one-way power family, then so is $\mathcal{F}'$. We now argue about the security of $\mathcal{F}'$, with an argument that works for any OWPF family.

Proposition 7.25. *If $\mathcal{F}$ is a family of $\Big(s(\lambda) + \mathsf{poly}(\lambda), \delta(\lambda)\Big)$-secure k-OWPFs, then for any non-constant $\lambda = \mathsf{poly}(n)$, it holds that $\mathcal{F}'$ is an $\Big(s(\lambda), \delta'(n)\Big)$-secure family of k-OWPFs, where $\delta'(n)$ is the maximum of:*

- $\delta(\lambda) \cdot \left(\frac{2^{-n}}{2^{-\lambda}}\right)^k$ *and*
- $k \cdot 2^{-\lambda} \cdot 2^{2n}$.

Proof. This follows by an argument almost identical to that of Proposition 7.23.

Let $P^{(n)}$ denote the distribution of $(H_1, \ldots, H_k, \mathbf{X})$ in the experiment defined by independently sampling $H_1, \ldots, H_k \leftarrow \mathcal{H}_n$ and $\mathbf{W} \leftarrow (\{0,1\}^n)^k$, and then defining $X_1 = H_1(W_1), \ldots, X_k = H_k(W_k)$. Specifically, we have

$$P^{(n)}(h_1, \ldots, h_k, \mathbf{x}) = \prod_{i=1}^{k} \Pr_{H_i \leftarrow \mathcal{H}_n}[H_i = h_i] \cdot \frac{\prod_{i=1}^{k} \left|\{w : h(w) = x_i\}\right|}{2^{kn}}. \tag{7.4}$$

Let $Q^{(n)}$ denote the distribution of $(H_1, \ldots, H_k, \mathbf{X})$ in the experiment defined by independently sampling $H_1, \ldots, H_k \leftarrow \mathcal{H}_n$ and $\mathbf{X} \leftarrow (\{0,1\}^\lambda)^k$. Specifically, we have

$$Q^{(n)}(h_1, \ldots, h_k, \mathbf{x}) = \prod_{i=1}^{k} \Pr_{H_i \leftarrow \mathcal{H}_n}[H_i = h_i] \cdot \Pr_{\mathbf{X} \leftarrow (\{0,1\}^\lambda)^k}[\mathbf{X} = \mathbf{x}]. \tag{7.5}$$

We first note that if $h_1, \ldots, h_k$ are all injective functions, then $P^{(n)}(h_1, \ldots, h_k, \mathbf{x}) \leq 2^{k\lambda} 2^{-kn} \cdot Q^{(n)}(h_1, \ldots, h_k, \mathbf{x})$ for all $\mathbf{x} \in \{0,1\}^\lambda$. We also note that by a union bound, the k hash functions $H_1, \ldots, H_k$ are all injective with probability at least $1 - k \cdot 2^{2n} 2^{-\lambda}$. Thus, the security of $\mathcal{F}'$ follows from the security of $\mathcal{F}$ by an identical reduction as in Proposition 7.23. □

Moreover, when $\mathcal{F} = \mathcal{G}^k$ is a one-way power family, then $\mathcal{F}'$ is also statistically injective with (essentially) the same parameters as in Claim 7.22.1. For each j, let $\textsc{Inj}_j$ denote the event that f'_{I_j, H_j} is injective, and let $\textsc{Inj} = \bigcup_j \textsc{Inj}_j$.

Claim 7.25.1. *Suppose that $\mathcal{F} = \mathcal{G}^k$ is a δ-secure one-way power family. Then,*

$$\Pr[\neg \textsc{Inj}] \leq k \cdot 2^{2n} \cdot \delta(\lambda)^{\frac{1}{k}}.$$

Proof. By symmetry, $\Pr[\textsc{Inj}_j]$ is independent of j, and moreover the $\textsc{Inj}_j$ are independent events. Therefore, we have

$$\Pr[\textsc{Inj}] \leq \sum_j \Pr[\textsc{Inj}_j] = k \prod_j \Pr[\textsc{Inj}_j]^{\frac{1}{k}} = k \Pr[\textsc{Inj}_1 \wedge \ldots \wedge \textsc{Inj}_k]^{\frac{1}{k}}.$$

But $\Pr[\textsc{Inj}_1 \wedge \ldots \wedge \textsc{Inj}_k]$ is at most $2^{2kn} \cdot \delta$ by the same reasoning as in Claim 7.22.1. Namely, $\Pr[\textsc{Inj}_1 \wedge \ldots \wedge \textsc{Inj}_k]$ is at most $2^{2kn} \cdot \mathsf{CP}_{[1:k]}$, where $\mathsf{CP}_{[1:k]}$ denotes the probability that a uniformly random k-tuple of pairs $((x_{1,1}, x_{1,2}), \ldots, (x_{k,1}, x_{k,2})) \in (\{0,1\}^\lambda)^{2k}$ satisfies $f_{I_j}(x_{j,1}) = f_{I_j}(x_{j,2})$ for every j; this follows from the pairwise independence of $\mathcal{H}_n$ and the fact that $H_1, \ldots, H_k$ are sampled independently.

Moreover, $\mathsf{CP}_{[1:k]}$ is at most $\delta(\lambda)$, as an adversary that on input $(I_1, \ldots, I_k, y_1, \ldots, y_k)$ guesses $x_1, \ldots, x_k$ uniformly at random succeeds in batch inverting $\mathcal{F}$ with probability at most δ, but succeeds with probability at least $\mathsf{CP}_{[1:k]}$. This completes the proof of Claim 7.25.1. □

Thus, we have a transformation from one-way power families to statistically injective one-way power families with essentially the same security loss as in the case of symmetric k-OWPFs.

On the other hand, for general OWPFs, we can only prove the following weaker claim about injectivity.

Claim 7.25.2. *Suppose that $\mathcal{F}$ is a δ-secure k-OWPF family. Then,* $\Pr[\neg\textsc{Inj}] \leq k \cdot 2^{2n} \cdot \delta \cdot 2^{(k-1)\lambda}$.

The parameters in this claim are tight for the following reason: suppose that $\mathcal{G}$ is a perfectly secure $(k-1)$-OWPF family and $\mathcal{F}$ is defined so that a member of $\mathcal{F}$ is a member of $\mathcal{G}$ combined with a constant function (as the kth function f_k). Then, no k-tuple of functions in $\mathcal{F}'$ consists of k injective functions.

Proof of Claim 7.25.2. We claim that for any fixed j, $\Pr[\neg\textsc{Inj}_j] \leq 2^{2n} \cdot \delta \cdot 2^{(k-1)\lambda}$; the desired result then follows from a union bound.

The fact that $\Pr[\neg\textsc{Inj}_j] \leq 2^{2n} \cdot \delta \cdot 2^{(k-1)n}$ follows by a similar argument to that of Claim 7.22.1. Namely, $\Pr[\neg\textsc{Inj}_j]$ is at most $2^{2n} \cdot \mathsf{CP}_j$, where CP_j denotes the probability that a random pair $(x_1, x_2) \in (\{0,1\}^\lambda)^2$ satisfies $f'_{j,I}(x_1) = f'_{j,I}(x_2)$; this follows from the pairwise independence of $\mathcal{H}_n$. But this probability in turn is at most $\delta \cdot 2^{(k-1)\lambda}$, as an adversary that on input $(I, y_1, \ldots, y_k)$ guesses $x_1, \ldots, x_k$ uniformly at random succeeds in batch inverting $\mathcal{F}$ with probability at most δ, but succeeds with probability at least $2^{-(k-1)\lambda} \cdot \mathsf{CP}_j$.

This completes the proof of Claim 7.25.2. □

7.3.4 From OWPFs to Symmetric OWPFs

In this section, we will construct families of symmetric OWPFs in two different ways: one construction is from general OWPF families, while the other is from one-way power families (Definition 7.19). The two reductions will have different security losses.

As usual, we assume that all functions in a fixed OWPF family have input domain $\{0,1\}^n$.

Theorem 7.26. *Let $\mathcal{F} = \{(f_{1,I}, \ldots, f_{k,I})\}_{I\in\mathcal{I}}$ be a $(s + \mathsf{poly}(n), \delta)$-secure family of k-OWPFs with domain $\{0,1\}^n$. Then, for any L, the function family $\mathcal{F}' = \{(f'_I, f'_I, \ldots, f'_I)\}_{I\in\mathcal{I}}$ is a (s, δ')-secure family of symmetric L-OWPFs, where $f'_I(x||j) = j||f_{j,I}(x)$ and*

$$\delta' = \delta + k(1 - \frac{1}{k})^L \min(\delta \cdot 2^{(k-1)n}, 1).$$

Remark 7.27. *Note that if all $f_{j,I}$ are injective with probability $1-\eta$, then a random element of the family $\mathcal{F}'$ is injective with probability at least $1-\eta$.*

Proof. Suppose that some size s adversary $\mathcal{A}(y'_1, \ldots, y'_L)$ wins the OWPF security game for $\mathcal{F}'$ with probability ϵ, where $y'_i = j_i||f_{j_i,I}(x_i)$ for each i. Let Win denote the event that $\mathcal{A}$ produces L valid inverses (i.e. it wins the security game), and let Distinct be the event that $\{j_1, \ldots, j_L\}$ contains at least k distinct elements. We prove two claims about the behavior of $\mathcal{A}$.

Claim 7.27.1. $\Pr[\textit{Win} \wedge \textit{Distinct}] \leq \delta$.

Proof. This follows from the $(s + \mathsf{poly}(n), \delta)$-security of $\mathcal{F}$. Namely, a k-OWPF adversary $\mathcal{A}'$ given $(\mathcal{I}, y_1, \ldots, y_k)$ can select $j_1, \ldots, j_L \overset{\$}{\leftarrow} [k]$ at random and prepare a L-OWPF challenge for $\mathcal{A}$ containing each y_i in a location t with $j_t = i$ (not including the challenge y_i if there is no such location). This perfectly simulates the OWPF security game for $\mathcal{A}$, and in the event that Win $\wedge$ Distinct occurs, $\mathcal{A}'$ obtains inverses to all k of its challenges. Thus, we conclude the claim by the security of $\mathcal{F}$. □

Claim 7.27.2. $\Pr[\textit{Win} \mid \neg\textit{Distinct}] \leq \delta \cdot 2^{(k-1)n}$.

Proof. This also follows from the $(s + \mathsf{poly}(n), \delta)$-security of $\mathcal{F}$. Namely, a k-OWPF adversary $\mathcal{A}'$ given $(\mathcal{I}, y_1, \ldots, y_k)$ can select $j_1, \ldots, j_L \overset{\$}{\leftarrow} [k]$ subject to the event $\neg$Distinct (this can be done efficiently) and prepare a L-OWPF challenge for $\mathcal{A}$ containing y_{j_1} in location 1. Whenever $\mathcal{A}$ successfully inverts its first challenge, $\mathcal{A}'$ can guess its other $k-1$ challenges uniformly at random and win with probability $2^{-(k-1)n}$. Thus, we conclude the claim by the security of $\mathcal{F}$. □

Finally, we note the combinatorial fact that $\Pr[\textsc{Distinct}] \le k(1-\frac{1}{k})^L$. Combining the two claims and this fact, we obtain the statement of Theorem 7.26. □

Remark 7.28. *Setting $L \approx k \log(\frac{1}{\delta}) < k^2 n$, we see that the family $\mathcal{F}'$ defined above is a $(s, \delta(1+o(1)))$-secure family of symmetric L-OWPFs.*

Remark 7.29. *If we instead set $k = 2$, $\log^{1.1}(n) < L < \frac{n}{\log^{1.1}(n)}$, and $\delta = 2^{-2n+\frac{L}{2}}$, we obtain a construction of $2^{-n-L\log(L)} \cdot \mathrm{negl}(n)$-secure symmetric L-OWPFs from suitably strong (asymmetric) 2-OWPFs. This is sufficient for L-multi-collision resistant hash functions (MCRHFs) if the original family $\mathcal{F}$ is also statistically injective. While this requires almost perfect security from the original OWPF family, we see this as a proof of concept that the most general notion of OWPF can be used without obfuscation to build more expressive primitives, such as MCRHFs.*

We now give a construction of symmetric OWPFs from one-way power families that has a milder security loss than the construction of Theorem 7.26; in the event that the one-way power family is *public coin*, the security loss can be improved even further.

Theorem 7.30. *Let $\mathcal{F}^k = \{(f_{I_1}, \ldots, f_{I_k})\}_{(I_1,\ldots,I_k)\in\mathcal{I}^k}$ be a public coin $(ks+\mathsf{poly}(n), \delta)$-secure one way k-power family with domain $\{0,1\}^n$. Moreover, for any $N = 2^{\nu(n)}$, and suppose that $\mathcal{H}$ is a family of programmable k-wise independent hash functions from $[N] \to \mathcal{I}$, where $\mathcal{I}$ is the key space for $\mathcal{F}$.[8] Then, for any L, the function family ${\mathcal{F}'}^L = \{(f'_h, f'_h, \ldots, f'_h)\}_{h\in\mathcal{H}}$ is a (s, δ')-secure family of L-OWPFs with domain*

[8] This is possible when either (1) $\mathcal{F}$ is public coin, or (2) $N = \mathsf{poly}(n,k)$, in which case sampling from $\mathcal{H}$ consists of sampling N independent keys from $\mathcal{I}$.

$\{0,1\}^{n+\nu(n)}$, where

$$f'_h(x||\rho) = \rho||f_{h(\rho)}(x)$$

and

$$\delta' = \delta + (k-1) \cdot \max_{1 \le d \le k-1} \left[\frac{d^d}{d!} \left(\frac{d}{N} \right)^{L-d} \delta^{\frac{1}{\lceil k/d \rceil}} \right].$$

In the special case that $N = \mathsf{poly}(n,k)$ and a member of the "hash family" $\mathcal{H}$ consists of N independently sampled $I_1, \ldots, I_N \in \mathcal{I}$, we obtain the same conclusion when $\mathcal{F}$ is not public coin.

Remark 7.31. *Note that if all members of the family $\mathcal{F}$ are injective, then f'_h is injective for every choice of hash function h.*

Proof. Suppose that some size s adversary $\mathcal{A}(h, y'_1, \ldots, y'_L)$ wins the OWPF security game for $\mathcal{F}'^k$ with probability ϵ, where $y'_i = \rho_i||f_{I_i}(x_i)$ and $I_i = \mathcal{F}.\mathsf{Samp}(\rho_i)$ for each i. Let Win denote the event that $\mathcal{A}$ produces L valid inverses (i.e. it wins the security game), and let d-Distinct be the event that $\{\rho_1, \ldots, \rho_L\}$ contains exactly d distinct elements. We prove the following claim about the behavior of $\mathcal{A}$.

Claim 7.31.1. $\Pr[\textit{Win} \mid d\textit{-Distinct}] \le \delta^{\frac{1}{\lceil k/d \rceil}}$.

Proof. This follows from a two-part argument. First, we note that the family $\mathcal{F}^d$ is a public coin $(s, \delta^{\frac{1}{\lceil k/d \rceil}})$-secure one-way d-power family. This is because any algorithm breaking $\mathcal{F}^d$ could be used $\lceil k/d \rceil$ times independently to break $\mathcal{F}^k$.

Thus, we prove the claim by reducing from the one-wayness of $\mathcal{F}^d$. In particular, an adversary $\mathcal{A}'$ given d independently drawn indices $I_1, \ldots, I_d$ and values $y_i = f_{I_i}(x_i)$ to invert could use $\mathcal{A}$ to break $\mathcal{F}^d$ in the following way.

- First, sample L uniformly random values $\rho_1, \ldots, \rho_L \leftarrow [N]$ such that there are exactly d distinct ρ_i. Call these values $\rho^*_1, \ldots, \rho^*_d$.

- Sample a hash function $h \leftarrow \mathcal{H}.\mathsf{CondGen}(\rho^*, \mathbf{I})$, i.e., a hash function subject to the constraints that $h(\rho^*_i) = I_i$. Here, we think of the indices I_i as public coins so that this sampling is possible.

- Run $\mathcal{A}(h, y'_1, \ldots, y'_L)$, where $y'_i = \rho_i || y_i$.

By the conditional sampling property of $\mathcal{H}$ (and k-wise independence), the input distribution to $\mathcal{A}$ in this experiment is exactly the correct input distribution (a random input subject to the constraint d-Distinct), so by the $\delta^{\frac{1}{\lceil k/d \rceil}}$-hardness of $\mathcal{F}^d$, we conclude the claim. □

Finally, we note that by a counting argument,

$$\Pr[d\text{-}\textsc{Distinct}] = \binom{N}{d}\left(\frac{d}{N}\right)^L \leq \frac{d^d}{d!}\left(\frac{d}{N}\right)^{L-d}.$$

Thus, we conclude Theorem 7.30 by a standard probability calculation.

□

Corollary 7.32. *Consider the case when $\mathcal{F}$ is public coin, $k = 2$ and $L = 3$, and set $\nu(n) = \frac{1}{4}\log(\frac{1}{\delta})$. Then, we have*

$$\delta' = \delta + \frac{1}{N^2}\delta^{\frac{1}{2}} = 2\delta,$$

with a new security parameter of $n' = n + \frac{1}{4}\log(\frac{1}{\delta})$. For example, this yields a $2^{-n'} \cdot \mathrm{negl}(n')$-secure symmetric 3-OWPF family from $2^{-4n/3} \cdot \mathrm{negl}(n)$-secure (public coin) 2-one-way power families (which suffice for 3-MCRHFs if $\mathcal{F}$ is also injective), and a $2^{(-4/3+o(1))n} \cdot \mathrm{negl}(n)$-secure symmetric 3-OWPF family from $2^{(-2+o(1))n}$-secure (public coin) one-way 2-power families.

Corollary 7.33. *Consider the case when $\mathcal{F}$ is public coin, $L = k(1+\log(n))$ and set $\nu(n) = \frac{n}{\log(n)}$. Then, we have*

$$\delta' < \delta + k^L 2^{-kn},$$

yielding essentially a (s, δ)-secure symmetric L-OWPF family from δ-secure (public coin) one-way k-power families. This reduction suffices for many of the applications in Section 7.5 if $\mathcal{F}$ is injective.

Corollary 7.34. *Consider the case $N = ek \cdot (nk)^c$; then, setting $L = k + \frac{1}{c\log(kn)}\log(\frac{1}{\delta})$,* *we obtain $\delta' < (1+o(1))\delta$. This yields L-MCRHFs from any statistically injective*[9]*, δ-secure one-way k-power families with $\delta < \left(2^{-n-k\log(k)}\right)^{\frac{1}{1-1/c}}$.*[10] *We therefore also obtain L-MCRHFs from sufficiently secure (not necessarily injective) one-way k-power families by first applying Construction 7.24 and then applying the construction of Theorem 7.30.*

7.4 Collision Resistance from OWPFs

Having defined and explored the foundations of OWPFs in Section 7.3, we now turn to *applications* of OWPFs. In this section, we prove our main theorem on collision resistance. As usual, we assume that all OWPFs used have domain $\{0,1\}^n$.

Theorem 7.35. *Suppose that $\mathcal{F}$ is an $(s(n), \delta(n))$-secure symmetric 2-OWPF-family $\mathcal{F}$ that is injective with probability $1-\eta$. Then, for every $m = m(n)$, the hash family $\mathcal{H} := \mathcal{H}_{\mathcal{F},n,m(n)}$ in Construction 7.36 is (s', δ')-collision-resistant for $s' = s + \mathsf{poly}(n)$ and $\delta' = \eta + \delta \cdot 2^{2n-m}$.*

Construction 7.36. *Given input and output lengths n and m, and a symmetric 2-OWPF-family $\mathcal{F} = \{(f_I, f_I)\}_{I\in\mathcal{I}}$ given by algorithms $(\mathcal{F}.\mathsf{Gen}, \mathcal{F}.\mathsf{Eval})$, define the hash family $\mathcal{H} = \mathcal{H}_{\mathcal{F},n,m}$ by $(\mathcal{H}.\mathsf{Gen}, \mathcal{H}.\mathsf{Eval})$ as follows. Let $\ell = \mathsf{poly}(\lambda)$ denote a bound on the output length of f_I for I in the support of $\mathcal{F}.\mathsf{Gen}(1^\lambda)$.*

$\mathcal{H}.\mathsf{Gen}$: *On input 1^λ sample $I \leftarrow \mathcal{F}.\mathsf{Gen}(1^\lambda)$, and sample $H_{\mathsf{out}} : \{0,1\}^\ell \to \{0,1\}^m$ from a programmable pairwise independent hash family. Output (I, H_{out}) as the hash function description.*

$\mathcal{H}.\mathsf{Eval}$: *On input $\big((I, H_{\mathsf{out}}), x\big)$, output $H_{\mathsf{out}}(f_I(x))$.*

Corollary 7.37 (Follows from Theorem 7.35 and Section 7.3.3)**.** *If there exists a $(\mathsf{poly}(n), 2^{-n}\cdot \mathrm{negl}(n))$-secure symmetric 2-OWPF family that is injective with probability $1-\mathrm{negl}(n)$, then collision resistant hash families exist. Also, if there exists*

[9] If $\mathcal{F}$ is statistically injective, then $\mathcal{F}'$ is statistically injective because with overwhelming probability, all N of the sampled function keys $I_1, \ldots, I_n$ will be injective.

[10] This follows from the inequality $\delta < 2^{-(n+\nu(n))}2^{-L\log(L)}$.

a $(\mathsf{poly}(n), 2^{-1.6n} \cdot \mathrm{negl}(n))$*-secure symmetric* 2*-OWPF family (with no injectivity hypothesis), then there exist collision-resistant hash families. Finally, if symmetric* 2*-OWPFs with* nearly optimal security *(i.e., security* $(s, 2^{-2n(1+o(1))}s^c)$*) exist, then CRHFs with nearly optimal security also exist.*

Informally, we will define $H_{i,h_{\mathsf{out}}}(x) := h_{\mathsf{out}}(f_i(x))$ for $i \in \mathcal{I}$ and $h_{\mathsf{out}} \in \mathcal{H}_{\mathsf{out}}$ and refer to Construction 7.36 as the "outer hash construction."

For any 2-OWPF family $\mathcal{F}$ and associated outer hash construction $\mathcal{H} = \mathcal{H}_{\mathcal{F},m}$, we first prove that it is hard to find a *certain type* of "outer" collisions in $\mathcal{H}$.

Definition 7.38 (Outer and Inner Collisions)**.** *Let* $\mathcal{F}$ *be a 2-OWPF family and* $\mathcal{H} = \mathcal{H}_{\mathcal{F},m}$ *be an associated outer hash construction. We say that* $(x_0, x_1) \in \{0,1\}^n$ *is an* outer collision *with respect to* (i, h_{out}) *if* $H_{i,h_{\mathsf{out}}}(x_0) = H_{i,h_{\mathsf{out}}}(x_1)$ *but* $f_i(x_0) \neq f_i(x_1)$*. We say that* (x_0, x_1) *is an* inner collision *if* $f_i(x_0) = f_i(x_1)$*.*

Our result is as follows.

Theorem 7.39 (Outer Hash Lemma)**.** *For any polynomial* $m(n)$*, there exists*[11] *a polynomial* $p(n)$ *such that for any* $\big(s(n) + p(n), \delta(n)\big)$*-secure family* $\mathcal{F}$ *of symmetric 2-OWPFs, it is* $\big(s(n), \delta(n) \cdot 2^{2n-m(n)}\big)$*-hard to find any* outer collision *in* $\mathcal{H}_{\mathcal{F},m}$*, given* $(I, H_{\mathsf{out}}) \leftarrow \mathcal{H}.\mathsf{Gen}(1^n)$*.*

Proof. Suppose for the sake of contradiction that there is an adversary $\mathcal{A} = \{\mathcal{A}_n\}$ that violates the $\big(s(n), \delta(n) \cdot 2^{2n-m(n)}\big)$-hardness of finding outer collisions for $\mathcal{H}_{f,m}$. That is, (1) the size of $\mathcal{A}_n$ is at most $s(n)$, and (2) for infinitely many n, the probability that (X_0, X_1) is an outer collision with respect to (I, H_{out}) is some $\epsilon(n) > \delta(n) \cdot 2^{2n-m(n)}$ in the probability space defined by sampling $(I, H_{\mathsf{out}}) \leftarrow \mathcal{H}.\mathsf{Gen}(1^n)$ and $(X_0, X_1) \leftarrow \mathcal{A}_n(I, H_{\mathsf{out}})$.

We let $\mathrm{Expt}_n^{(0)}$ and $\mathrm{Pr}_n^{(0)}$ respectively denote the experiment described above and the probability measure that it induces. In $\mathrm{Expt}^{(0)}$, let $\textsc{Win}$ denote the event that (X_0, X_1) is an outer collision with respect to (I, H_{out}). We now define a sequence of re-

[11] In fact, $p(n) = \mathsf{poly}(n, m(n))$ for some polynomial poly that depends only on the programmable pairwise hash family $\mathcal{H}_{\mathsf{out}}$.

lated probability experiments $\left\{\mathrm{Expt}_n^{(j)}\right\}_{j\in\{1,2,3\}}$, and let $\{\Pr_n^{(j)}\}$ denote the probability measures that they induce.

- Let $\mathrm{Expt}_n^{(1)}$ denote the following modification of $\mathrm{Expt}_n^{(0)}$:

  1. Sample $(I, H_{\mathsf{out}}) \leftarrow \mathcal{H}.\mathsf{Gen}(1^n)$
  2. Sample $X_0^*, X_1^* \overset{\text{i.i.d.}}{\leftarrow} \{0,1\}^n$.
  3. Compute $(X_0, X_1) \leftarrow \mathcal{A}_n(I, H_{\mathsf{out}})$.

  In $\mathrm{Expt}_n^{(1)}$, let $\textsc{Win}$ denote the event that (X_0, X_1) is an outer collision with respect to (I, H_{out}). It holds that

$$\Pr_n^{(1)}\left[\textsc{Win} \wedge \left((X_0, X_1) = (X_0^*, X_1^*)\right)\right] = \Pr_n^{(0)}\left[\textsc{Win}\right] \cdot 2^{-2n} \geq \frac{\epsilon(n)}{2^{2n}}.$$

- Let $\mathrm{Expt}_n^{(2)}$ denote the following further modification.

  1. Sample (I, X_0^*, X_1^*) as in $\mathrm{Expt}_n^{(1)}$. If $f_I(X_0^*) = f_I(X_1^*)$, abort.
  2. Sample $Z_0^*, Z_1^* \overset{\text{i.i.d.}}{\leftarrow} \{0,1\}^{m(n)}$.
  3. Sample $H_{\mathsf{out}} \leftarrow \mathcal{H}.\mathsf{CondGen}(\mathbf{Y}^*, \mathbf{Z}^*)$, where $\mathbf{Y}^* = (f_I(X_0^*), f_I(X_1^*))$ and $\mathbf{Z}^* = (Z_0^*, Z_1^*)$.
  4. Compute $(X_0, X_1) \leftarrow \mathcal{A}_n(I, H_{\mathsf{out}})$.

  In $\mathrm{Expt}_n^{(2)}$, let $\textsc{Win}$ denote the event that (1) $f_I(X_0^*) \neq f_I(X_1^*)$ (so that the experiment proceeds to completion) and (2) (X_0, X_1) is an outer collision with respect to (I, H_{out}). Then, it holds that

$$\Pr_n^{(2)}\left[\textsc{Win} \wedge \left((X_0, X_1) = (X_0^*, X_1^*)\right)\right] = \Pr_n^{(1)}\left[\textsc{Win} \wedge \left((X_0, X_1) = (X_0^*, X_1^*)\right)\right] \geq \frac{\epsilon}{2^{2n}}.$$

  by the pairwise independence and programmability of $\mathcal{H}_{\mathsf{out}}$ (and the fact that $\mathbf{Z}^*$ was chosen uniformly at random).

- Let $\mathrm{Expt}^{(3)}$ denote the following further modification.

1. Sample (I, X_0^*, X_1^*) as in $\mathrm{Expt}^{(1)}$. If $f_I(X_0^*) = f_I(X_1^*)$, abort.
2. Sample $Z^* \leftarrow \{0,1\}^{m(n)}$ and define $Z_0^* = Z_1^* = Z^*$.
3. Sample $H_{\mathsf{out}} \leftarrow \mathcal{H}.\mathsf{CondGen}(\mathbf{Y}^*, \mathbf{Z}^*)$, where $\mathbf{Y}^* = (f_I(X_0^*), f_I(X_1^*))$ and $\mathbf{Z}^* = (Z_0^*, Z_1^*)$.
4. Compute $(X_0, X_1) \leftarrow \mathcal{A}_n(I, H_{\mathsf{out}})$.

In $\mathrm{Expt}_n^{(3)}$, let the event $\textsc{Win}$ be defined as in $\mathrm{Expt}_n^{(2)}$. Then

$$\begin{aligned}
\Pr_n^{(3)}\left[(X_0, X_1) = (X_0^*, X_1^*)\right] &\geq \Pr_n^{(3)}\left[\textsc{Win} \wedge (X_0, X_1) = (X_0^*, X_1^*)\right] \\
&= \Pr_n^{(2)}\left[\textsc{Win} \wedge (X_0, X_1) = (X_0^*, X_1^*) | Z_0^* = Z_1^*\right] \\
&= \frac{\Pr_n^{(2)}\left[\textsc{Win} \wedge (X_0, X_1) = (X_0^*, X_1^*)\right]}{\Pr_n^{(2)}\left[Z_0^* = Z_1^*\right]} \qquad (7.6) \\
&\geq \frac{\epsilon \cdot 2^{-2n}}{2^{-m(n)}} = \epsilon \cdot 2^{m(n)-2n}. \qquad (7.7)
\end{aligned}$$

where Eq. (7.6) follows because the event "$\textsc{Win} \wedge (X_0, X_1) = (X_0^*, X_1^*)$" occurs *only* when $Z_0^* = Z_1^*$.

We now deduce the existence of an $\left(s(n) + \mathsf{poly}(n), \epsilon(n) \cdot 2^{m(n)-2n}\right)$-attack on the 2-OWPF security of $\mathcal{F}$. The attack is given by the following algorithm $\mathcal{B} = \{\mathcal{B}_n\}$. On input (I, Y_0^*, Y_1^*), $\mathcal{B}_n$ does the following:

1. Sample $Z^* \leftarrow \{0,1\}^{m(n)}$
2. Sample $H_{\mathsf{out}} \leftarrow \mathcal{H}_{\mathsf{out}}.\mathsf{CondGen}(\mathbf{Y}^*, (Z^*, Z^*))$
3. Compute and output $(X_0, X_1) \leftarrow \mathcal{A}_n(I, H_{\mathsf{out}})$.

Suppose that as in the 2-OWPF security game, $\mathcal{B}_n$'s input (I, Y_0^*, Y_1^*) is generated by sampling $I \leftarrow \mathcal{F}.\mathsf{Gen}(1^n)$; $X_0^*, X_1^* \overset{\text{i.i.d.}}{\leftarrow} \{0,1\}^n$; and $Y_b^* = f_I(X_b^*)$ for each $b \in \{0,1\}$. Then all of our named random variables are jointly distributed exactly as in $\mathrm{Expt}_n^{(3)}$. Thus the output (X_0, X_1) of $\mathcal{B}_n$ is equal to (X_0^*, X_1^*) (and in particular $\mathcal{B}_n$ has inverted both Y_0^* and Y_1^*) with probability at least $\epsilon(n) \cdot 2^{m(n)-2n} > \delta(n)$.

This concludes the proof of Theorem 7.39. □

Finally, we give a proof of Theorem 7.35.

Proof of Theorem 7.35. Suppose there is some size s adversary $\mathcal{A}$ that on input (I, H_{out}) outputs $x_1 \neq x_2$ such that $H_{I,H_{\mathsf{out}}}(x_1) = H_{I,H_{\mathsf{out}}}(x_2)$ with probability δ'; that is, $\mathcal{A}$ finds a collision with this probability. Note that with probability $1 - \eta$ over the randomness of $\mathcal{H}.\mathsf{Gen}$, *no inner collisions exist* in $H_{I,H_{\mathsf{out}}}$. Moreover, note that by Theorem 7.39, $\mathcal{A}$ outputs an outer collision with probability at most $\delta \cdot 2^{2n-m}$. We conclude Theorem 7.35 by a union bound. □

7.4.1 Parameter Settings and Discussion

When we aim for polynomially-secure CRHFs from $\{0,1\}^n \to \{0,1\}^m$, the 2-OWPF assumption required by Theorem 7.35– namely, $2^{m-2n} \cdot \text{negl}(n)$-secure injective symmetric 2-OWPFs – is plausible for any $m = \omega(\log(n))$.

We also obtain "optimally hard" collision resistant hash functions under plausible assumptions (which, for example, are satisfied by our "double discrete logarithm" candidate). The relevant result is sketched in Corollary 7.37, but to be more specific, our hash function is collision-resistant with $2^{-m(1-\epsilon)}$-security assuming the existence of a $2^{-2n+\epsilon m} \cdot \text{negl}(n)$-secure (injective symmetric) 2-OWPF, which is plausible for any $m = \omega(\frac{\log(n)}{\epsilon})$. This yields (for any super-logarithmic output length) a collision resistant hash family with security nearly matching the trivial attack of outputting two uniformly random points x_1, x_2. Moreover, by Section 7.3.3, the injectivity requirement on the 2-OWPF family can be removed (with slightly more security loss).

In terms of optimality, we recall that by Theorem 7.7 (see Section 7.7), the construction of CRHFs from $2^{-n} \cdot \text{negl}(n)$-secure injective symmetric 2-OWPFs cannot be quantitatively improved (with black box techniques); indeed, even one-way permutations with security $\frac{2^{-n}}{\text{negl}(n)}$ do not imply CRHFs in a black-box way. Thus, constructing $2^{-n} \cdot \text{negl}(n)$-secure injective symmetric 2-OWPFs from $2^{-\frac{n}{2}} \cdot \text{negl}(n)$-secure one-way permutations (or even $2^{-.99n}$-secure one-way permutations) is an extremely interesting open question.

As a final note on collision resistance, recall that Corollary 7.37 shows that CRHFs exist as long as sufficiently secure *symmetric* 2-OWPFs exist (without having to assume injectivity), but none of our OWPF transformations currently suffice to build CRHFs from asymmetric OWPFs. We leave the question of whether CRHFs can be constructed from (sufficiently secure) arbitrary 2-OWPFs open.

7.5 Output Intractability from OWPFs

In this section, we generalize the proof strategy of Section 7.4 to build correlation intractable hash functions for all k-ary output relations ("k-output intractable hash functions") assuming suitably secure k-OWPF families exist. The hardness that we need depends quantitatively on the *sparsity* of the relation R.

We now define the relevant objects and assumptions for our construction.

Definition 7.40 (Correlation Intractability)**.** *A hash family $\mathcal{H} = \{h_I\}_{I\in\mathcal{I}}$ (as in Definition 7.11) is said to be (s, δ)-*multi-input correlation intractable *for a class $\mathcal{R}$ of relations if for every $2k$-ary relation $R \in \mathcal{R}$ and every size-$s(\cdot)$ circuit ensemble $\{\mathcal{A}_n\}$,*

$$\Pr_{\substack{I\leftarrow \mathsf{Gen}(1^\lambda)\\ (x_1,\ldots,x_k)\leftarrow\mathcal{A}_\lambda(I)}}[(x_1,\ldots,x_k,h_I(x_1),\ldots,h_I(x_k))\in R] \le O(\delta(\lambda)).$$

*Additionally, $\mathcal{H}$ is said to be δ-*multi-input correlation intractable *with respect to $\mathcal{R}$ if $\mathcal{H}$ is (n^c, δ) multi-input correlation intractable for every $c > 0$, and $\mathcal{H}$ is said to be* multi-input correlation intractable *with respect to $\mathcal{R}$ if $\mathcal{H}$ is $(n^c, m(n)^{-c})$ multi-input correlation intractable for every $c > 0$.*

Correlation intractability is a useful and versatile property of random oracles that we would like to guarantee in the standard model. However, even a random oracle $\mathcal{O}$ is not correlation intractable with respect to relations R whose accepting inputs are sufficiently dense. To avoid this problem, we restrict our relations R to be *sparse* as in Definition 7.41 below.

Definition 7.41 (Sparsity)**.** *For any relation $R \subseteq (\{0,1\}^*)^k \times (\{0,1\}^*)^k$, we say that*

*R is $p(\cdot)$-*sparse *if for any $\mathbf{x} \in (\{0,1\}^*)^k$,*

$$\Pr_{\mathbf{y} \leftarrow (\{0,1\}^m)^k} [(\mathbf{x}, \mathbf{y}) \in R] \leq p(m).$$

When p is a negligible function, we say simply that R is sparse*.*

Ideally, we would construct a hash family that is correlation intractable for all sparse relations. However, our OWPF-based construction is only able to handle k-ary relations R that depend only on the *outputs* of $\mathcal{H}$ rather than the inputs.

Definition 7.42 (Output Intractability)**.** *We say that a hash family $\mathcal{H}$ is (s, δ)-*output intractable *for a class $\mathcal{R}$ of relations if $\mathcal{H}$ is (s, δ)-multi-input correlation intractable for $\mathcal{R}$, and every relation in $\mathcal{R}$ (1) requires that $x_1, \ldots, x_k$ are distinct, and (2) is otherwise only a function of the* outputs *y_i of $\mathcal{H}$ (and not the inputs).*

We note that requiring distinct inputs $x_1, \ldots, x_k$ is necessary in order for our notion of sparsity to be applicable; this is because the random variable $(H(x_1), \ldots, H(x_k))_{H \leftarrow \mathcal{H}}$ cannot be a uniformly random k-tuple if $x_i = x_j$ for some $i \neq j$. However, every k-ary relation $R(y_1, \ldots, y_k)$ can be thought of as a union of at most k^k relations to which Definition 7.42 can be applied.

Moreover, we note that in Section 7.6.3, we are able to construct hash functions that go beyond output intractability, at the cost of introducing indistinguishability obfuscation as an additional assumption.

Finally, we discuss the notion of *samplability* of a relation, which will prove useful in our security proof.

Definition 7.43 (t-Samplability of a relation R)**.** *An output relation $R \subseteq (\{0,1\}^m)^k$ is samplable in time t if there is a sampling algorithm S such that (1) $S(1^n, 1^k)$ runs in time $t = t(n)$, and (2) for every $\mathbf{y} \in R$,*

$$\Pr[S(1^n, 1^k) = y_i \textit{ for all } i] = \Pr_{\mathbf{Y} \leftarrow (\{0,1\}^m)^k}[\mathbf{Y} = \mathbf{y} \mid R(\mathbf{Y}) = 1].$$

In other words, the distribution sampled by $S(1^n, 1^k)$ is the uniform distribution on the set of $\mathbf{y}$ for which $R(\mathbf{y}) = 1$.

We say that R is efficiently samplable if it is samplable in time $\mathsf{poly}(n,k)$.

Remark 7.44. *Any output relation $R \subseteq (\{0,1\}^m)^k$ is samplable by a* non-uniform *algorithm running in time $t = 2^{2km} \cdot \mathsf{poly}(m)$ by enumerating over all outputs $(y_1, \dots, y_k)$, computing each $R(y_1, \dots, y_k)$ (using a circuit of size 2^{km}), and selecting a uniformly random k-tuple out of those satisfying R.*

Let $\mathcal{R}^{\text{out}}_{k,p,t}$ denote the class of k-ary output relations

$$R = \{R_n \subseteq (\{0,1\}^n)^{k(n)} \times (\{0,1\}^m)^{k(n)}\}$$

that are p-sparse and samplable in time t,[12] and let $\mathcal{R}^{\text{out}}_{k,p} = \bigcup_t \mathcal{R}^{\text{out}}_{k,p,t}$ denote the class of k-ary output relations that are p-sparse. For any $R \in \mathcal{R}^{\text{out}}_{k,p}$, we will abuse notation and think of R as both a relation on $(\{0,1\}^m)^k$ (i.e. the outputs) and a relation on $(\{0,1\}^n)^k \times (\{0,1\}^m)^k$ (the output relation along with the constraint that the inputs x_i are all distinct).

We now state our results on output intractability.

Theorem 7.45. *Suppose that $\mathcal{F} = \{f_I : \{0,1\}^n \to \{0,1\}^{\ell(n)}\}_{I \in \mathcal{I}}\}$ is a family of symmetric k-OWPFs with security $(s + \mathsf{poly}(n), \delta)$, and suppose further that $\mathcal{F}$ is injective with probability $1 - \eta$. For every $m = m(n)$, let $\mathcal{H} := \mathcal{H}_{\mathcal{F},n,m(n)}$ denote the hash family in Construction 7.36. Then, for every sparsity p, $\mathcal{H}$ is (s, δ')-output intractable for $\mathcal{R}^{\text{out}}_{k,p}$ with $\delta' = \eta + \delta \cdot p \cdot 2^{kn}$.*

Moreover, if a relation $R \in \mathcal{R}^{\text{out}}_{k,p,t}$ is samplable in uniform time t, then there is a uniform *reduction to OWPF security with an additional loss of t time.*

Construction 7.46. *Suppose we are given a symmetric k-OWPF family $\mathcal{F} = \{f_I\}$ with input space $\{0,1\}^n$ and output space $\{0,1\}^{\ell(n)}$ given by algorithms $(\mathcal{F}.\mathsf{Gen}, \mathcal{F}.\mathsf{Eval})$. We define the hash family $\mathcal{H} = \mathcal{H}_{\mathcal{F},n,m}$ by $(\mathcal{H}.\mathsf{Gen}, \mathcal{H}.\mathsf{Eval})$ as follows.*

$\mathcal{H}.\mathsf{Gen}$: *On input 1^λ sample $I \leftarrow \mathcal{F}.\mathsf{Gen}(1^\lambda)$, and sample $H_{\mathsf{out}} : \{0,1\}^\ell \to \{0,1\}^m$ from a programmable k-wise independent hash family $\mathcal{H}_{\mathsf{out}}$. Output (I, H_{out}) as the hash function description.*

[12] We may consider both uniform and non-uniform versions of this definition.

$\mathcal{H}.\mathsf{Eval}$: *On input* $\big((I, H_{\mathsf{out}}), x\big)$, *output* $H_{\mathsf{out}}(\mathcal{F}.\mathsf{Eval}(I, x))$.

Remark 7.47. *For various parameter settings, Theorem 7.45 can be combined with the reductions of Section 7.3.3 and Section 7.3.4 to obtain constructions from certain asymmetric and/or non-injective OWPFs. See Proposition 7.23 and Section 7.3.4 for some examples.*

Informally, we will define $H_{I,h_{\mathsf{out}}}(x) := h_{\mathsf{out}}(f_I(x))$ for $I \in \mathcal{I}$ and $h_{\mathsf{out}} \in \mathcal{H}_{\mathsf{out}}$ and call Construction 7.46 the "(generalized) outer hash construction."

We will prove Theorem 7.45 by generalizing the outer hash lemma (Theorem 7.39) to the case of general output intractability. That is, for any k-OWPF family $\mathcal{F}$ and associated outer hash construction $\mathcal{H} = \mathcal{H}_{\mathcal{F},n,m}$, and for any output relation $R \in \mathcal{R}^{\mathrm{out}}_{k,p,t}$, we prove that $\mathcal{H}$ is correlation intractable with respect to a *modified* relation $R_{\mathcal{F}}$:

Definition 7.48 (Post-Composed Relation R_f)**.** *For any output relation* $R \subseteq (\{0,1\}^m)^k$ *and any function* $f : \{0,1\}^n \to \{0,1\}^{\ell(n)}$*, we define the* post-composed relation R_f *by*

$$R_f(\mathbf{x}, \mathbf{y}) = 1 \textit{ if and only if } R(\mathbf{y}) = 1 \textit{ and } f(x_1), \ldots, f(x_k) \textit{ are distinct.}$$

In the case of collision resistance, this definition corresponds to the notion of an "outer collision."

Our result is as follows.

Theorem 7.49 (Generalized Outer Hash Lemma)**.** *Let* t' *be the runtime of the sampling algorithm* $\mathcal{H}_{\mathsf{out}}.\mathsf{CondGen}$*. If* $\mathcal{F}$ *is a* $(s+t'+t, \delta)$*-secure* k*-OWPF against uniform adversaries and* $R \in \mathcal{R}^{\mathrm{out}}_{k,p,t}$*, then* $\mathcal{H}$ *is* $(s, \delta \cdot \frac{2^{kn}}{p})$*-correlation intractable with respect to* $R_{\mathcal{F}}$ *(i.e. the relation* R_{f_I} *depends on the hash function* $(I, \mathcal{H}_{\mathsf{out}}) \leftarrow \mathcal{H}.\mathsf{Gen}$*). Moreover, the same conclusion holds if* $\mathcal{F}$ *is a* $(s + t', \delta)$*-secure* k*-OWPF with respect to nonuniform adversaries.*

Proof. Suppose that some s-time adversary $\mathcal{A}$, on input $(I, H_{\mathsf{out}}) \leftarrow \mathcal{H}.\mathsf{Gen}(1^n)$, produces with probability ϵ an input $\mathbf{x}$ such that $R_{f_I}(\mathbf{x}, \mathbf{y}) = 1$, where $y_i = H_{I,H_{\mathsf{out}}}(x_i)$ for each i. Let the random variable $\mathbf{X} = (X_1, \ldots, X_k)$ denote the output of $\mathcal{A}(I, H_{\mathsf{out}})$, let $Y_i = H_{I,H_{\mathsf{out}}}(X_i)$ for all i, and let the random variable $\textsc{Win}$ denote the event that $R_{f_I}(\mathbf{X}, \mathbf{Y}) = 1$. We will call $\mathrm{Expt}^{(0)}$ the security game described above.

- Consider the following modified experiment $\mathrm{Expt}^{(1)}$. A challenger generates $(I, H_{\mathsf{out}}) \leftarrow \mathcal{H}.\mathsf{Gen}(1^n)$, chooses uniformly random $\mathbf{X}^* \overset{\$}{\leftarrow} (\{0,1\}^n)^k$, and sends (I, H_{out}) to $\mathcal{A}$, which in turn outputs $\mathbf{X}$. Then, we have

$$\Pr^{(1)}\left[\textsc{Win} \wedge (\mathbf{X} = \mathbf{X}^*)\right] = \Pr^{(0)}\left[\textsc{Win}\right] \cdot 2^{-kn} \geq \frac{\epsilon}{2^{kn}}.$$

  We note that if the variables $Y_i^* := f_I(X_i^*)$ are not distinct in $\mathrm{Expt}^{(1)}$ then $\mathcal{A}$ necessarily loses, so we redefine the game to immediately end if this occurs.

- Consider the further modified experiment $\mathrm{Expt}^{(2)}$, defined as follows. The challenger generates $(I, \mathbf{X}^*)$ as above, and additionally generates $\mathbf{Z}^* \overset{\$}{\leftarrow} (\{0,1\}^m)^k$ uniformly at random. The challenger then samples $H_{\mathsf{out}} \leftarrow \mathcal{H}.\mathsf{CondGen}(\mathbf{Y}^*, \mathbf{Z}^*)$ and sends (I, H_{out}) to $\mathcal{A}$. Then, we have

$$\Pr^{(2)}\left[\textsc{Win} \wedge (\mathbf{X} = \mathbf{X}^*)\right] = \Pr^{(1)}\left[\textsc{Win} \wedge (\mathbf{X} = \mathbf{X}^*)\right] \geq \frac{\epsilon}{2^{kn}}$$

  by the programmability correctness of $\mathcal{H}_{\mathsf{out}}$ (and the fact that $\mathbf{Z}^*$ was chosen uniformly at random).

- Consider an experiment $\mathrm{Expt}^{(3)}$ which differs from $\mathrm{Expt}^{(2)}$ only in that $\mathbf{Z}^*$ is instead sampled by $S(1^n, 1^k)$, the sampling algorithm associated to R. Then

$$\begin{aligned}
\Pr^{(3)}\left[\mathbf{X} = \mathbf{X}^*\right] &\geq \Pr^{(3)}\left[\textsc{Win} \wedge \mathbf{X} = \mathbf{X}^*\right] \\
&= \Pr^{(2)}\left[\textsc{Win} \wedge \mathbf{X} = \mathbf{X}^* | R(\mathbf{Z}^*) = 1\right] && (7.8) \\
&= \frac{\Pr^{(2)}\left[\textsc{Win} \wedge \mathbf{X} = \mathbf{X}^*\right]}{\Pr^{(2)}\left[R(\mathbf{Z}^*) = 1\right]} && (7.9) \\
&\geq \frac{\epsilon \cdot 2^{-kn}}{p}. && (7.10)
\end{aligned}$$

  where Eq. (7.8) follows from our correctness requirement of the sampling algorithm S, Eq. (7.9) follows because the event "$\textsc{Win} \wedge \mathbf{X} = \mathbf{X}^*$" occurs *only* when $R(\mathbf{Z}^*) = 1$, and Eq. (7.10) follows from the p-sparsity of R.

- Finally, we note that $\mathrm{Expt}^{(3)}$ leads to a $(s + t + t', \epsilon \cdot \frac{2^{-kn}}{p})$-attack on the k-OWPF security of $\mathcal{F}$. The attack is as follows: an adversary $\mathcal{A}'$ given $(I, \mathbf{Y}^*)$ as in the k-OWPF security game can sample $\mathbf{Z}^* \leftarrow S(1^n, 1^k)$, sample $H_{\mathsf{out}} \leftarrow \mathcal{H}_{\mathsf{out}}.\mathsf{CondGen}(\mathbf{Y}^*, \mathbf{Z}^*)$, and run $\mathcal{A}(I, H_{\mathsf{out}})$. This perfectly simulates $\mathrm{Expt}^{(3)}$, and hence $\mathcal{A}'$ recovers $\mathbf{X}^*$ (which in particular satisfies $f_I(X_i^*) = Y_i^*$ for all i) with probability at least $\epsilon \cdot \frac{2^{-kn}}{p}$.

As a uniform algorithm, the OWPF adversary $\mathcal{A}'$ runs in time $s + t' + t$, but since the sampling step $\mathbf{Z}^* \leftarrow S(1^n, 1^k)$ is oblivious to the OWPF challenge $\mathbf{Y}^*$, by an averaging argument there *exists* some string $\mathbf{z}^* \in (\{0,1\}^m)^k$ such that $\mathcal{A}'$ with $\mathbf{Z}^* := \mathbf{z}^*$ hardwired also inverts $\mathbf{Y}^*$ with probability $\epsilon \cdot \frac{2^{-kn}}{p}$, which yields a nonuniform attack running in time $s + t'$. This concludes the proof of Theorem 7.49. □

Finally, we give a proof of Theorem 7.45.

Proof of Theorem 7.45. Suppose there is some size s adversary $\mathcal{A}$ that on input (I, H_{out}) outputs $\mathbf{x}$ such that all x_i are distinct and $R(\mathbf{y}) = 1$, where $y_i = H_{I,H_{\mathsf{out}}}(x_i)$ for all i, with probability δ'. Note that with probability $1 - \eta$ over the randomness of $\mathcal{H}.\mathsf{Gen}$, the function f_I is injective, so by a union bound, $\mathcal{A}$ wins its security game *and* f_I is injective with probability at least $\delta' - \eta$. However, if $\mathcal{A}$ wins its security game and f_I is injective, then $\mathcal{A}$ has produced an input $\mathbf{x}$ such that $R_{f_I}(\mathbf{x}, \mathbf{y}) = 1$. But by Theorem 7.49, this can happen with probability at most $\delta \cdot 2^{kn} \cdot p$. Thus, we conclude that $\delta' \leq \eta + \delta \cdot 2^{kn} \cdot p$, as desired. □

7.5.1 Examples Arising from Theorem 7.45

We now we describe some of the consequences of Theorem 7.45 for particular relations R of interest.

Collision Resistance

As a direct consequence of Theorem 7.45, we recover our theorem on collision resistance, Theorem 7.35. The relevant output relation is defined as follows: $R(y_1, y_2) = 1$ if and only if $y_1 = y_2$. R has sparsity 2^{-m}, where m is the output length of our hash function $H : \{0,1\}^n \to \{0,1\}^m$. Moreover, the set $\{\mathbf{y} : R(\mathbf{y}) = 1\}$ is exactly $\{(r,r) : r \in \{0,1\}^m\}$ and is therefore polynomial-time samplable (meaning that we do not have to rely on a non-uniform reduction).

Thus, we recover Theorem 7.35 from Theorem 7.45, as the distinguishing advantage δ produced by Theorem 7.45 is $\delta = 2^{-2n} \cdot 2^m \cdot \mathrm{negl}(n)$ for any negligible function $\mathrm{negl}(n)$.

Multi-Collision Resistance

By considering the k-ary output relation

$$R(\mathbf{y}) = 1 \text{ if and only if } y_1 = y_2 = \ldots = y_k,$$

we obtain a result on k-collision resistance [KNY17,BDRV18,BKP18,KNY18] for any k. This relation has sparsity $2^{-(k-1)m}$ (if the hash function has output length m), and we can efficiently sample a random $\mathbf{y}$ such that $R(\mathbf{y}) = 1$ by choosing a uniformly random $r \leftarrow \{0,1\}^m$ and outputting $(r, \ldots, r)$. Thus, by Theorem 7.45, we have the following result.

Corollary 7.50. *If there exists an injective symmetric OWPF family with security (s, δ), then there exists a family of k-MCRHFs mapping $\{0,1\}^n \to \{0,1\}^m$ with security roughly $(s, \delta \cdot 2^{kn} \cdot 2^{-(k-1)m})$. (Moreover, this is proved by a uniform reduction.)*

In particular, for any $m = \omega(\log n)$, a plausible setting of δ yields a k-collision resistant hash family whose security matches the trivial attack of outputting k uniformly random points $x_1, \ldots, x_k$.

Finally, we consider the special case $m = n - \log(k)$ (the minimal compression to guarantee k-collisions) and polynomial security, in which case we require an injective

OWPF that is $2^{-n-k\log(k)} \cdot \mathrm{negl}(n)$-secure. For example, in the case of $k = \frac{\alpha n}{\log(n)}$, we require $2^{-(1+\alpha)n}$-hardness of a problem for which the naive algorithm has success probability $2^{-\frac{\alpha^2 n^2}{\log^2(n)}}$. This is a substantially weaker OWPF assumption than is required for collision resistance.

We note that by [BDRV18, KNY18], k-collision-resistant hash functions (for any k) suffice to build constant-round statistically-hiding commitments, another primitive which we currently do not know how to construct from IO and one-way functions alone.

Additionally, the quantitatively weaker (injective symmetric) OWPF requirements for k-MCRHFs allow us to use reductions from both Section 7.3.3 and Section 7.3.4 to obtain constructions from various kinds of asymmetric and/or non-injective OWPFs. We refer the reader to these previous sections for details.

7.6 Constructions from IO and OWPFs

In this section, we combine OWPFs with the powerful notion of *indistinguishability obfuscation* in the hopes of obtaining better constructions of hash functions. We successfully obtain:

- A better quantitative tradeoff than in constructions based on general (i.e. *asymmetric*) OWPFs, avoiding a costly intermediate reduction such as Theorem 7.26. For example, we obtain a construction of CRHFs from IO and $2^{-n}\cdot\mathrm{negl}(n)$-secure injective 2-OWPFs (without any symmetry requirement).

- A hash family that is correlation intractable with respect to a broader class of relations than achievable with (symmetric) OWPFs alone. As described later, this includes an instantiation of the Fiat-Shamir transform for an expressive class of interactive proofs.

Moreover, our constuction is extremely simple: our hash function is an obfuscated (puncturable) PRF $\mathcal{O}(F_s(\cdot))$, and we only require the existence of OWPFs in the security proofs. As a byproduct, this construction confirms our intuition that obfuscated (puncturable) PRFs should satisfy many random oracle properties (including collision-resistance, despite the negative result of [AS15]). Our work in this section extends the proof technique of [KRR17], who show that an obfuscated puncturable PRF suffices for Fiat-Shamir assuming the existence of strong point function obfuscation.

7.6.1 Preliminaries

Indistinguishability Obfuscation

An *obfuscator for all circuits* is a ppt algorithm $\mathcal{O}$ such that for every circuit C, $\mathcal{O}(C)$ is with probability 1 a circuit $\tilde{C}$ with the same functionality as C. Various security properties may be defined for an obfuscator; the one most relevant to us is *indistinguishability obfuscation* [BGI+01].

Definition 7.51 (Indistinguishability Obfuscation). *$\mathcal{O}$ is a (s,δ)-secure* indistinguishability obfuscator *(IO) if for all pairs of functionally equivalent circuits C_0 and C_1 of size $|C_0| = |C_1| = \lambda$, and all circuits $\mathcal{A}$ of size $s(\lambda)$, it holds that*

$$\Pr[\mathcal{A}(\mathcal{O}(C_0)) = 1] - \Pr[\mathcal{A}(\mathcal{O}(C_1)) = 1] \le O(\delta(\lambda)).$$

Puncturable PRFs

Definition 7.52 (Puncturable PRF [BW13,BGI14,KPTZ13,SW14]). *A PPRF family is a family of functions*

$$\mathcal{F} = \left\{ F_{n,s} : \{0,1\}^n \to \{0,1\}^{m(n)} \right\}_{n\in\mathbb{N}, s\in\{0,1\}^{\ell(n)}}$$

with associated (deterministic) polynomial-time algorithms $(\mathcal{F}.\mathsf{Eval}, \mathcal{F}.\mathsf{Puncture}, \mathcal{F}.\mathsf{PuncEval})$ *satisfying*

- *For all $x \in \{0,1\}^n$ and all $s \in \{0,1\}^{\ell(n)}$, $\mathcal{F}.\mathsf{Eval}(s,x) = F_{n,s}(x)$.*
- *For all distinct $x, x' \in \{0,1\}^n$ and all $s \in \{0,1\}^{\ell(n)}$, $\mathcal{F}.\mathsf{PuncEval}(\mathcal{F}.\mathsf{Puncture}(s,x), x') = \mathcal{F}.\mathsf{Eval}(s,x')$.*

For ease of notation, we write $F_s(x)$ and $\mathcal{F}.\mathsf{Eval}(s,x)$ interchangeably, and we write $s\{x\}$ to denote $\mathcal{F}.\mathsf{Puncture}(s,x)$.

$\mathcal{F}$ is said to be (s,δ)-secure if for every $\{x^{(n)} \in \{0,1\}^n\}_{n\in\mathbb{N}}$, the following two distribution ensembles (indexed by n) are $\delta(n)$-indistinguishable to circuits of size $s(n)$:

$$(S\{x^{(n)}\}, F_S(x^{(n)})) \text{ where } S \leftarrow \{0,1\}^{\ell(n)}$$

and

$$(S\{x^{(n)}\}, U) \text{ where } S \leftarrow \{0,1\}^{\ell(n)},\ U \leftarrow \{0,1\}^{m(n)}.$$

Theorem 7.53 ([GGM84, KPTZ13, BW13, BGI14, SW14])**.** *If {polynomially secure, subexponentially secure, subexponential advantage-secure} one-way functions exist, then for all functions $m : \mathbb{N} \to \mathbb{N}$ (with $1^{m(n)}$ polynomial-time computable from 1^n), and all $\delta : \mathbb{N} \to [0,1]$ with $\delta(n) \geq 2^{-\mathsf{poly}(n)}$, there is a polynomial $\ell(n)$ and a {polynomially secure, $(\frac{1}{\delta},\delta)$-secure, δ-secure} PPRF family*

$$\mathcal{F}_m = \left\{F_{n,s} : \{0,1\}^n \to \{0,1\}^{m(n)}\right\}_{n\in\mathbb{N}, s\in\{0,1\}^{\ell(n)}}.$$

7.6.2 Warm-Up: Target Collision Resistance

To demonstrate the power of our technique, we first show that an obfuscated PPRF $\mathcal{O}(F_s)$ is target collision-resistant (i.e. a UOWHF), only making use of the additional assumption that injective one-way functions exist. This result may be of independent interest – although one-way functions imply UOWHFs without additional assumptions [Rom90], we are not aware of any prior proof that $\mathcal{O}(F_s)$ (with suitable padding) is a UOWHF.[13] This result also demonstrates that the planting technique can be used without making any exponential assumptions.

Theorem 7.54. *Let $m : \mathbb{N} \to \mathbb{N}$ be a polynomial time computable function such that $n > m(n) \geq n - O(\log n)$. Suppose that*

- *$\mathcal{O}$ is a sub-exponential advantage-secure indistinguishability obfuscator.*

[13] In contrast, a standard puncturing argument suffices to prove that $\mathcal{O}(G \circ F'_s)$ *is* target collision-resistant, where G denotes a PRG and F'_s denotes a PRF with output length $\frac{m}{2}$.

- *$\mathcal{F} = \left\{ \{F_{n,s} : \{0,1\}^n \to \{0,1\}^{m(n)}\}_{s \in \{0,1\}^{\ell(n)}} \right\}_{n \in \{0,1\}^*}$ is a family of 2^{-2n}-secure puncturable PRFs. We will use the notation $F_s(\cdot)$ as shorthand.*
- *There* exists *a family $\mathcal{F}_{inj}$ of (polynomially secure) injective one-way functions.*

Then, there is a polynomial $p : \mathbb{N} \to \mathbb{N}$ such that the hash family $\mathcal{H}$ defined by $H \leftarrow \mathcal{O}(P_s)$ is a UOWHF family, where P_s is a program padded to have size $p(n)$ which on input $x \in \{0,1\}^n$ outputs $F_s(x)$.

Proof Overview We will show that if an adversary $\mathcal{A}$ finds collisions in H with noticeable probability, then it also finds a random *planted* collision in H with noticeable probability. On the other hand, we hide the planted collision with an special-purpose obfuscator (based on any injective one-way function), which exactly prevents $\mathcal{A}$ from finding the planted collision with noticeable probability.

Proof. The polynomial $p(n)$ is chosen to be large enough so that $\mathcal{O}$ is 2^{-2n}-secure for programs of length $p(n)$, and so that all circuits obfuscated in our proof's hybrids have size at most $p(n)$ (in particular, $p(n)$ must be at least as large as the description of a function in $\mathcal{F}_{\text{inj}}$).

Suppose that $\mathcal{H}$ is not a UOWHF – namely, for some ppt $(\mathcal{A}_0, \mathcal{A}_1)$, some $c > 0$, and infinitely many n, in the experiment $\text{Expt}^{(0)}$ defined by sampling $(X, \mathsf{st}) \leftarrow \mathcal{A}_0(1^n)$, $S \leftarrow \{0,1\}^{\ell(n)}$, $H \leftarrow \mathcal{O}(P_S)$ and $X' := \mathcal{A}_1(H, \mathsf{st})$, it holds that

$$\Pr^{(0)}[\textsc{Win}] > m(n)^{-c} := m^{-c},$$

where $\textsc{Win}$ denotes the event that $X \neq X'$ but $H(X) = H(X')$.

- Consider an experiment $\text{Expt}^{(1)}$ which differs from $\text{Expt}^{(0)}$ only in that we additionally (and independently) sample $X^* \leftarrow \{0,1\}^n$. Then clearly

$$\Pr^{(1)}[\textsc{Win} \wedge (X' = X^*)] = \Pr^{(0)}[\textsc{Win}] \cdot 2^{-n} > \frac{1}{2^n m^c}.$$

- Consider an experiment $\text{Expt}^{(2)}$ which differs from $\text{Expt}^{(1)}$ only in the definition of H. Namely, H is defined not as $\mathcal{O}(P_S)$, but as $\mathcal{O}(P_{S,X^*,F_S(X^*)})$, where P_{s,x^*,y^*}

is the appropriately padded circuit (with $s\{x^*\}$, x^*, and y^* hard-coded) that computes

$$P_{s,x^*,y^*}(x) = \begin{cases} y^* & \text{if } x = x^* \\ \mathsf{PuncEval}(s\{x^*\}, x) & \text{otherwise.} \end{cases}$$

Because $P_{S,X^*,F_s(X^*)}$ is functionally equivalent to P_S, the 2^{-2n} security of $\mathcal{O}$ implies that

$$\Pr^{(2)}\left[\textsc{Win} \wedge (X' = X^*)\right] \geq \Pr^{(1)}\left[\textsc{Win} \wedge (X' = X^*)\right] - 2^{-2n} > \frac{1}{2^n m^c} - 2^{-2n}.$$

- Consider an experiment $\mathrm{Expt}^{(3)}$ which differs from $\mathrm{Expt}^{(2)}$ only in the definition of H. Namely, H is now sampled as $\mathcal{O}(P_{S,X^*,Y^*})$ for independently and uniformly random $Y^* \leftarrow \{0,1\}^m$. Now the 2^{-2n} punctured pseudorandomness of F_s at X^* implies that

$$\Pr^{(3)}\left[\textsc{Win} \wedge (X' = X^*)\right] \geq \Pr^{(2)}\left[\textsc{Win} \wedge (X' = X^*)\right] - 2^{-2n} > \frac{1}{2^n m^c} - 2 \cdot 2^{-2n}.$$

- Consider an experiment $\mathrm{Expt}^{(4)}$ which differs from $\mathrm{Expt}^{(3)}$ only in that Y^* is now defined as $Y^* := F_S(X)$. Then

$$\begin{aligned} \Pr^{(4)}\left[X' = X^*\right] &\geq \Pr^{(4)}\left[\textsc{Win} \wedge (X' = X^*)\right] \\ &= \Pr^{(3)}\left[\textsc{Win} \wedge (X' = X^*) | Y^* = F_s(X)\right] \\ &= \frac{\Pr^{(3)}\left[\textsc{Win} \wedge (X' = X^*)\right]}{\Pr^{(3)}\left[Y^* = F_s(X)\right]} \qquad (7.11) \\ &> \left(\frac{1}{2^n m^c} - 2 \cdot 2^{-2n}\right) 2^m \geq \frac{1}{m^c \cdot 2^{O(\log(n))}} = \text{non-negl}(n), \end{aligned}$$

where Eq. (7.11) follows because the event "$\textsc{Win} \wedge (X' = X^*)$" occurs *only* when $Y^* = F_S(X)$.

- Finally, consider an experiment $\mathrm{Expt}^{(5)}$ which differs from $\mathrm{Expt}^{(4)}$ only in that H is now sampled as $\mathcal{O}(\tilde{P}_{S,f_I(X^*),Y^*})$, where $f_I \leftarrow \mathcal{F}_{\text{inj}}$ is sampled from the family of injective one-way functions, and $\tilde{P}_{s,w^*,y^*}$ is the circuit (with s, w^*, and

y^* hard-coded) that computes

$$\tilde{P}_{s,w^*,y^*}(x) = \begin{cases} y^* & \text{if } f_I(x) = w^* \\ F_s(x) & \text{otherwise.} \end{cases}$$

Since f_I is injective, we know that $\tilde{P}_{S,f_I(X^*),Y^*}$ is functionally equivalent to P_{S,X^*,Y^*}. We then have that $\Pr^{(5)}[X' = X^*] = \text{non-negl}(n)$ by the security of $\mathcal{O}$.

- However, this constitutes a polynomial-time inversion attack on $\mathcal{F}_{\text{inj}}$. Even if $\mathcal{A}$ were given $\tilde{P}_{S,f_{\text{inj}}(X^*),Y^*}$ in the clear, $\mathcal{A}$ should be unable to produce an inverse to $f_{\text{inj}}(X^*)$, as X^* is uniformly random and independent of S and Y^*. This contradicts the one-wayness of the family $\mathcal{F}_{\text{inj}}$, and so we have proved that $\mathcal{H}$ is a UOWHF. □

7.6.3 Multi-Input Correlation Intractability

In this section, we generalize the proof strategy of Section 7.6.2 to build multi-input correlation intractable hash functions – for a special class of relations that we define below – assuming the existence of IO, puncturable PRFs, and suitably secure injective k-OWPF families. The hardness that we need depends quantitatively on the *sparsity* of the relation R. Our proof relies on the observation that injective k-OWPFs allow us to obfuscate programs of the form

$$P_{x_1,\ldots,x_k}(x) = \begin{cases} i & x = x_i \text{ for some } i \\ 0 & \text{otherwise.} \end{cases}$$

Moreover, by combining our result here with Construction 7.24, we obtain a construction from suitably secure (asymmetric and non-injective) k-OWPFs.

We refer the reader to Section 7.5 for the relevant definitions about correlation intractability. We again note that ideally, we would prove that an obfuscated (puncturable) PRF is correlation intractable for all sparse relations. Indeed, our proof reduces correlation intractability for any sparse R to the *existence* of an (extremely secure) special-purpose obfuscator that depends on R.[14] When R is a k-ary relation that satisfies a "local sampleability" property, we construct such an obfuscator from injective OWPFs.

Definition 7.55 (Local Approximate Sampling with Setup). *A relation $R \subseteq (\{0,1\}^n)^k \times (\{0,1\}^m)^k$ is locally ϵ-approximately samplable with t-setup if there are k polynomial time algorithms $S_1, S_2, \ldots, S_k$ and a probabilistic algorithm* Setup *such that:*

- $\mathsf{Setup}(1^n, 1^k)$ *runs in time t and outputs a string* CRS *of length* $\mathsf{poly}(n,k)$.
- *For every $(\mathbf{x}, \mathbf{y}) \in R$,*

$$\Pr_{\mathsf{CRS}}[S_i(x_i; \mathsf{CRS}) = y_i \text{ for all } i] \geq \epsilon \cdot \Pr_{\mathbf{Y} \leftarrow (\{0,1\}^m)^k}[\mathbf{Y} = \mathbf{y} \mid R(\mathbf{x}, \mathbf{Y}) = 1].$$

  In other words, for every $\mathbf{x}$, the distribution $(S_i(x_i; \mathsf{CRS}))_i$ approximates the uniform distribution on the set of $\mathbf{y}$ for which $R(\mathbf{x}, \mathbf{y}) = 1$ as long as this set is non-empty.

We further restrict our attention to "distinct-input" relations, as we do in Section 7.5.

Definition 7.56 (Distinct-Input Relation). *A k-ary relation $R \in (\{0,1\}^n)^k \times (\{0,1\}^m)^k$ is a* distinct-input relation *if $R(\mathbf{x}, \mathbf{y}) = 0$ whenever $x_i = x_j$ for some $i \neq j \in [k]$.*

Let $\mathcal{R}_{k,t,\epsilon,p}$ denote the class of distinct-input k-ary relations

$$R = \{R_n \subseteq (\{0,1\}^n)^{k(n)} \times (\{0,1\}^m)^{k(n)}\}$$

that are p-sparse and locally ϵ-approximately samplable with t-setup, and let $\mathcal{R}_{k,\epsilon,p} := \bigcup_t \mathcal{R}_{k,t,\epsilon,p}$ denote the class of distinct-input k-ary relations that are p-sparse and locally ϵ-approximately samplable (with any setup time).

We now state our most general result on multi-input correlation intractability.

[14] In general, it is not clear when such obfuscators exist, and upon which assumptions they can be based.

Theorem 7.57. *Let $\nu : \mathbb{N} \to \mathbb{R}$ be a function satisfying $\nu(n) \geq 2^{-\mathsf{poly}(n)}$, let $k : \mathbb{N} \to \mathbb{N}$ be any polynomial, let $T, t : \mathbb{N} \to \mathbb{N}$ satisfy $t(n) \leq 2^{\mathsf{poly}(n)}$ and $T(n) \leq k(n) \cdot 2^n$. Suppose also that*

- *$\mathcal{O}$ is a sub-exponentially secure[15] indistinguishability obfuscator.*
- *$\mathcal{F} = \left\{ \{F_{n,s} : \{0,1\}^n \to \{0,1\}^{m(n)}\}_{s \in \{0,1\}^{\ell(n)}} \right\}_{n \in \mathbb{N}}$ is a family of $(2^{2kn}, \nu(m(n)) \cdot 2^{-2kn})$-secure puncturable PRFs. We will use the notation $F_s(\cdot)$ as shorthand for $F_{n,s}(\cdot)$.*
- *There* exists *a $(T + \mathsf{poly}(n), \delta)$-secure injective k-OWPF family $\mathcal{F}_{inj}$ for some $\delta = 2^{-kn} \cdot \frac{\epsilon}{p} \cdot \nu(m)$.*

Then, there is a polynomial $p : \mathbb{N} \to \mathbb{N}$ such that the hash family $\mathcal{H}$ defined by $H \leftarrow \mathcal{O}(P_s)$ is $(T, \nu(m(\cdot)))$-correlation intractable for $\mathcal{R}_{k,\epsilon,p}$, where P_s is a circuit that evaluates F_s (padded to size $p(n)$).

Moreover, for the restricted class $\mathcal{R}_{k,t,\epsilon,p}$, the reduction to OWPF security can be made uniform with an additional loss of t time.

Remark 7.58. *The restriction to distinct-input relations is primarily for ease of presentation; in particular, any $2k$-ary relation R is a union of at most k^k distinct-input relations, so at the cost of parameters that are worse by a factor of k^k, Theorem 7.57 can be applied to sparse relations not necessarily satisfying the distinct-input condition.*

Proof. The polynomial $p(n)$ is chosen to be large enough so that $\mathcal{O}$ is $(t+2^{2kn}, \nu(m(n)) \cdot 2^{-2kn} \cdot \epsilon)$-secure for programs of length $p(n)$, and so that all circuits obfuscated in our proof's hybrids have size at most $p(n)$.

Let R be any relation in $\mathcal{R}_{k,t,\epsilon,p}$, and suppose that an adversary $\mathcal{A}$ breaks the $(T, \nu(m(\cdot)))$-correlation intractability of $\mathcal{H}$ for R. We define $\mathrm{Expt}^{(0)}$ to be the R-correlation intractability game: $S \leftarrow \{0,1\}^{\ell(n)}$, $H \leftarrow \mathcal{O}(P_S)$, and $\mathbf{X} := (X_1, \ldots, X_k) \leftarrow$

[15] Correlation intractability for any fixed relation R can be achieved from a potentially weaker assumption; $\mathcal{O}$ and $\mathcal{F}$ must be secure against circuits of size that depends on t and the time to decide R.

$\mathcal{A}(H)$. Moreover, we define $\mathbf{Y} := Y_1||\dots||Y_k := H(X_1)||\dots||H(X_k)$, and define WIN to be the event that $R(\mathbf{X}, \mathbf{Y}) = 1$. We then argue as follows.

- Consider an experiment $\mathrm{Expt}^{(1)}$ which differs from $\mathrm{Expt}^{(0)}$ only in that we additionally (and independently) sample $\mathbf{X}^* := (X_1^*, \dots, X_k^*) \leftarrow (\{0,1\}^n)^k$. Then,

$$\Pr^{(1)}[\textsc{Win} \wedge (\mathbf{X} = \mathbf{X}^*)] = \Pr^{(0)}[\textsc{Win}] \cdot 2^{-kn} > \omega(\nu(m)) \cdot 2^{-kn}.$$

  Note that when $(X_1^*, \dots, X_i^*)$ are not distinct in $\mathrm{Expt}^{(1)}$, $\mathcal{A}$ necessarily loses, so we re-define the game to immediately end if this event occurs.

- Consider an experiment $\mathrm{Expt}^{(2)}$ which differs from $\mathrm{Expt}^{(1)}$ only in the definition of H. Namely, H is sampled not as $\mathcal{O}(P_S)$, but as $\mathcal{O}(P_{S,\mathbf{X}^*,\mathbf{Y}^*})$, where $\mathbf{Y}^* := (Y_1^*, \dots, Y_k^*) \leftarrow (\{0,1\}^m)^k$ is drawn uniformly at random, and $P_{s,\mathbf{x}^*,\mathbf{y}^*}$ is the appropriately padded circuit (with s, $\mathbf{x}^*$, and $\mathbf{y}^*$ hard-coded) that computes

$$P_{s,\mathbf{x}^*,\mathbf{y}^*}(x) = \begin{cases} y_1^* & \text{if } x = x_1^* \\ \vdots & \vdots \\ y_k^* & \text{if } x = x_k^* \\ F_s(x) & \text{otherwise.} \end{cases}$$

  Then, we have that

$$\begin{aligned} \Pr^{(2)}[\textsc{Win} \wedge \mathbf{X} = \mathbf{X}^*] &\geq \Pr^{(1)}[\textsc{Win} \wedge \mathbf{X} = \mathbf{X}^*] - O(k \cdot 2^{-2kn}) \\ &> \frac{\omega(\nu(m))}{2^{kn}} - O(k \cdot \nu(m) \cdot 2^{-2kn}) = \frac{\omega(\nu(m))}{2^{kn}}. \end{aligned}$$

  where we have invoked the $(2^{2kn}, \nu(m(n)) \cdot 2^{-2kn})$ security[16] of $\mathcal{O}$ ($k+1$ times) and the $(2^{2kn}, \nu(m(n)) \cdot 2^{-2kn})$ security of $\mathcal{F}$ (k times) to puncture the program P_S at each X_i^*.

- Consider an experiment $\mathrm{Expt}^{(3)}$ which differs from $\mathrm{Expt}^{(2)}$ only in how $\mathbf{Y}^*$ is

[16] This level of security is required because determining whether WIN occurs requires deciding R.

sampled. Specifically, conditioned on $\mathbf{X}^* = \mathbf{x}^*$, its distribution is uniform on $\{\mathbf{y} \in (\{0,1\}^m)^k : R(\mathbf{x}^*, \mathbf{y}) = 1\}$ whenever this set is non-empty. Then,

$$\begin{aligned}
\Pr^{(3)}[\mathbf{X} = \mathbf{X}^*] &\geq \Pr^{(3)}[\textsc{Win} \wedge \mathbf{X} = \mathbf{X}^*] \\
&= 2^{-kn} \sum_{\mathbf{x}^*} \Pr^{(3)}[\textsc{Win} \wedge \mathbf{X} = \mathbf{X}^* | \mathbf{X}^* = \mathbf{x}^*] \\
&= 2^{-kn} \sum_{\mathbf{x}^*} \Pr^{(2)}[\textsc{Win} \wedge R(\mathbf{x}^*, \mathbf{Y}^*) = 1 \wedge \mathbf{X} = \mathbf{X}^* | \mathbf{X}^* = \mathbf{x}^*] \\
&= 2^{-kn} \sum_{\mathbf{x}^*} \frac{\Pr^{(2)}[\textsc{Win} \wedge \mathbf{X} = \mathbf{X}^* \mid \mathbf{X}^* = \mathbf{x}^*]}{\Pr^{(2)}[R(\mathbf{x}^*, \mathbf{Y}^*) = 1]} \qquad (7.12) \\
&\geq 2^{-kn} \sum_{\mathbf{x}^*} \frac{\Pr^{(2)}[\textsc{Win} \wedge \mathbf{X} = \mathbf{X}^* \mid \mathbf{X}^* = \mathbf{x}^*]}{p} \qquad (7.13) \\
&= \frac{\Pr^{(2)}[\textsc{Win} \wedge \mathbf{X} = \mathbf{X}^*]}{p} = \frac{\omega(\nu(m))}{2^{kn} \cdot p}
\end{aligned}$$

where Eq. (7.12) follows because the event "$\textsc{Win} \wedge \mathbf{X} = \mathbf{x}^*$" occurs *only* when $R(\mathbf{x}^*, \mathbf{Y}^*) = 1$, and Eq. (7.13) follows from the p-sparsity of R.

- Consider an experiment $\mathrm{Expt}^{(4)}$ which differs from $\mathrm{Expt}^{(3)}$ only in how $\mathbf{Y}^*$ is sampled. Specifically, conditioned on $\mathbf{X}^* = \mathbf{x}^*$, $\mathbf{Y}^*$ is equal to $(S_i(x_i^*, \mathsf{CRS}))_{i=1}^k$, where $\mathsf{CRS} \leftarrow \mathsf{Setup}(1^n || 1^k)\}$. Then,

$$\begin{aligned}
\Pr^{(4)}[\mathbf{X} = \mathbf{X}^*] &= \sum_{\mathbf{x}^*, \mathbf{y}^*} \Pr^{(4)}\Big[\mathbf{X} = \mathbf{X}^* \mid (\mathbf{X}^*, \mathbf{Y}^*) = (\mathbf{x}^*, \mathbf{y}^*)\Big] \Pr^{(4)}\Big[(\mathbf{X}^*, \mathbf{Y}^*) = (\mathbf{x}^*, \mathbf{y}^*)\Big] \\
&= 2^{-kn} \sum_{\mathbf{x}^*, \mathbf{y}^*} \Pr^{(4)}\Big[\mathbf{X} = \mathbf{X}^* \mid (\mathbf{X}^*, \mathbf{Y}^*) = (\mathbf{x}^*, \mathbf{y}^*)\Big] \Pr^{(4)}\Big[\mathbf{Y}^* = \mathbf{y}^* \mid \mathbf{X}^* = \mathbf{x}^*\Big] \\
&\geq \epsilon \cdot 2^{-kn} \sum_{\mathbf{x}^*, \mathbf{y}^*} \Pr^{(3)}\Big[\mathbf{X} = \mathbf{X}^* \mid (\mathbf{X}^*, \mathbf{Y}^*) = (\mathbf{x}^*, \mathbf{y}^*)\Big] \Pr^{(3)}\Big[\mathbf{Y}^* = \mathbf{y}^* \mid \mathbf{X}^* = \mathbf{x}^*\Big] \\
& \qquad (7.14) \\
&= \epsilon \cdot \Pr^{(3)}[\mathbf{X} = \mathbf{X}^*] = \frac{\epsilon \cdot \omega(\nu(m))}{2^{kn} \cdot p}
\end{aligned}$$

where Eq. (7.14) follows from the approximate sampling condition for $(S_1, \ldots, S_k)$ and the fact that $\mathrm{Expt}^{(4)}$ and $\mathrm{Expt}^{(3)}$ only differ in the sampling of $\mathbf{y}^*$.

- Finally, consider an experiment $\mathrm{Expt}^{(5)}$ which differs from $\mathrm{Expt}^{(4)}$ only in that

H is now sampled as $\mathcal{O}(\tilde{P}_{S,\mathbf{W}^*,\mathsf{CRS}})$, where $W_i^* := f_i(X_i^*)$, $(f_1, \dots, f_k) \leftarrow \mathcal{F}_{\mathrm{inj}}$ is sampled from the OWPF family, and $\tilde{P}_{s,\mathbf{w}^*,\mathsf{crs}}$ is the circuit (with s, $\mathbf{w}^*$, and crs hard-coded) that computes

$$\tilde{P}_{s,\mathbf{w}^*,\mathsf{crs}}(x) = \begin{cases} S_i(x;\mathsf{crs}) & \text{if } f_i(x) = w_i^* \text{ for some } i \\ F_s(x) & \text{otherwise.} \end{cases}$$

Since the f_i are all injective, we know that $\tilde{P}_{S,\mathbf{W}^*,\mathsf{crs}}$ is functionally equivalent to $P_{S,\mathbf{X}^*,\mathbf{Y}^*}$ for $\mathbf{Y}^* = (S_i(X_i^*;\mathsf{CRS}))_{i=1}^k$. We then have that $\Pr^{(5)}[\mathbf{X} = \mathbf{X}^*] = \frac{\epsilon\cdot\omega(\nu(m))}{2^{kn}\cdot p}$ by the $(T + t + \mathsf{poly}(n), \epsilon \cdot \nu(m) \cdot 2^{-2kn})$-security of $\mathcal{O}$.

- However, the adversary's success in $\mathrm{Expt}^{(5)}$ contradicts the $(t + \mathsf{poly}(n), 2^{-kn} \cdot \frac{\epsilon}{p} \cdot \nu(m))$-security of $\mathcal{F}_{\mathrm{inj}}$. In particular, a modified adversary $\mathcal{B}$ given only $W_i^* := f_i(X_i^*)$ for all i could sample $\tilde{P}_{S,\mathbf{W}^*,\mathsf{CRS}}$ itself in time $t + \mathsf{poly}(n)$ and feed this output to $\mathcal{A}$, solving the batch inversion problem with probability $\frac{\epsilon\cdot\omega(\nu(m))}{2^{kn}\cdot p}$. This constitutes a $(T + t + \mathsf{poly}(n), \frac{\epsilon\cdot\omega(\nu(m))}{2^{kn}\cdot p})$ attack on the OWPF family, which completes the claimed uniform reduction. Moreover, we note that the CRS sampling algorithm $\mathsf{Setup}(1^n, 1^k)$ is oblivious to the OWPF challenge, so by an averaging argument there *exists* some string crs such that $\mathcal{B}$ with $\mathsf{CRS} := \mathsf{crs}$ hardcoded wins the OWPF security game with the same probability. This completes the nonuniform reduction, proving correlation intractability for every $R \in \mathcal{R}_{k,\epsilon,p}$. □

7.6.4 Examples Arising from Theorem 7.57

We now we describe some of the consequences of Theorem 7.57 for particular relations R of interest.

Collision Resistance

As a direct consequence of Theorem 7.57, we obtain a second construction of collision-resistant hash functions. Similarly to before, the relevant relation is defined as follows:

$R(x_1, x_2, y_1, y_2) = 1$ if and only if $x_1 \neq x_2$ and $y_1 = y_2$. As noted in Section 7.5.1, R has sparsity 2^{-m}, and the set $\{\mathbf{y} : R(\mathbf{x}, \mathbf{y}) = 1\}$ is efficiently sampleable in a way that is *oblivious* to the input $\mathbf{x}$. Thus R is clearly locally 1-sampleable with polynomial-time setup.

Thus, we obtain the following corollary.

Corollary 7.59. *If $\mathcal{O}$ is a sub-exponential advantage-secure indistinguishability obfuscator, $\mathcal{F}$ is a sub-exponential advantage-secure puncturable PRF, and there exists a δ-secure injective 2-OWPF family, then an $\mathcal{O}$-obfuscation of a (sufficiently padded) PRF chosen from $\mathcal{F}$ is $\delta \cdot 2^{2n} \cdot 2^{-m}$- collision resistant (by a uniform reduction).*

This exactly matches the quantitative parameters of Theorem 7.35. However, there are significant differences between the two results, namely:

- Corollary 7.59 requires the existence of sub-exponential advantage-secure IO, but

- Corollary 7.59 only requires (injective) OWPFs rather than *symmetric* (injective) OWPFs. Moreover, Corollary 7.59 only requires that such OWPFs exist; they are not required in the construction itself. Theorem 7.39, even when combined with the reductions of Section 7.3, was unable to produce a construction of CRHFs from (injective) asymmetric OWPFs.

Since the quantitative parameters of Corollary 7.59 match those of Theorem 7.35, this also yields CRHFs with optimal security under plausible OWPF assumptions (and IO).

Output Intractability

We also obtain an analog to Theorem 7.45; that is, a result on output intractability.

Corollary 7.60. *If $\mathcal{O}$ is a sub-exponentially secure indistinguishability obfuscator, $\mathcal{F}$ is a sub-exponentially secure puncturable PRF, and there exists a δ-secure injective k-OWPF family, then a $\mathcal{O}$-obfuscation of a (sufficiently padded) PRF chosen from*

$\mathcal{F}$ is $\delta \cdot 2^{kn} \cdot p$-output intractable for all k-ary output relations R. Moreover, this reduction can be made uniform (with a time t loss) if R is t-samplable.

Again, this involves the same quantitative OWPF parameters as in Theorem 7.45, with the same tradeoff as in the collision resistance example above.

An Example Falling Outside the Output Intractability Framework

All of our previous examples are special cases of output intractability as defined in [Zha16] (albeit with possibly unbounded relations, unlike [Zha16]). On the other hand, consider the following relation on $(\{0,1\}^n)^2 \times (\{0,1\}^m)^2$, parametrized by a matrix $\mathbf{A} \in \mathbb{F}_2^{m\times n}$:

$$R_{\mathbf{A}}(x_1, x_2, y_1, y_2) = 1 \text{ if } x_1 \neq x_2 \text{ and } y_1 \oplus y_2 = \mathbf{A}(x_1 \oplus x_2).$$

This is clearly not a special case of output intractability (the relation depends explicitly on both the inputs and outputs). However, it falls into the framework captured by Theorem 7.57. The relation $R_{\mathbf{A}}$ has sparsity 2^{-m}. We can also sample, for any $x_1 \neq x_2 \in \{0,1\}^m$, a random (y_1, y_2) such that $R_{\mathbf{A}}(x_1, x_2, y_1, y_2) = 1$ with the algorithms

$$S_i(x_i; r) = r \oplus \mathbf{A}x_i.$$

Thus, we see that an obfuscated PRF is correlation intractable for these relations assuming an injective 2-OWPF family with the exact same parameters as those required for collision resistance.

In fact, this example extends to the following relation on $(\{0,1\}^n)^2 \times (\{0,1\}^m)^2$, parametrized by a matrix $\mathbf{A} \in \mathbb{F}_2^{d\times n}$ and a full-rank matrix $\mathbf{B} \in \mathbb{F}_2^{d\times m}$, as long as $2^{-d} = \text{negl}(n)$:

$$R_{\mathbf{A},\mathbf{B}}(x_1, x_2, y_1, y_2) = 1 \text{ if } x_1 \neq x_2 \text{ and } \mathbf{B}(y_1 \oplus y_2) = \mathbf{A}(x_1 \oplus x_2).$$

The Fiat-Shamir Transform for Commit-Challenge-Response Proofs

Theorem 7.57 is also applicable in the case $k = 1$: we give new sufficient conditions for the provably secure instantiation of the Fiat-Shamir heuristic [FS87], for an expressive class of interactive proof systems. Namely, we consider the familiar example of "commit-challenge-response" proofs.

Definition 7.61 (Commit-Challenge-Response Proof System)**.** *A 3-message proof system* $\Pi = (P, V)$ *is called a* commit-challenge-response *proof system for a language* L *if it satisfies the following properties.*

1. *The first message is sent by the prover to the verifier. This message, which we denote by* a*, consists of a block-wise commitment (under a statistically binding commitment scheme) to a string* y *that is a function of both the common input* x *and the prover's private input* w*.*

2. *The second message, which we denote by* e *and refer to as the verifier's "challenge", is sent by the verifier to the prover and is sampled uniformly at random from a* $\mathsf{poly}(\lambda)$*-size alphabet* Σ*.*

3. *The third and final message, which we denote by* z*, is sent by the prover to the verifier, and consists of a* decommitment *to* y_T*, i.e., a subset* T *of the blocks of* y*. Here,* T *is a function of the challenge* e*.*

4. *The verifier* V *accepts if and only if (1)* z *is a valid decommitment of* a_T*, and (2) the tuple* (x, y_T, e) *passes some efficient test* Check*, where* y_T *is the value to which* a_T *was decommitted.*

Examples of commit-challenge-response proof systems include the classical 3-message zero knowledge protocol for 3-coloring [GMW86] as well as the 3-message zero knowledge protocol for Hamiltonicity given by [FLS90] (with a slight modification).

As we will see shortly, it is possible to use Theorem 7.57 to instantiate the Fiat-Shamir heuristic for any commit-challenge-response protocol (repeated in parallel).

The key advantage to using our approach over that of [KRR17], or the more recent work of [CCRR18], is that we prove security only assuming that IO and exponentially secure one-way functions exist, rather than needing (exponentially secure) input-hiding obfuscation for arbitrary multi-bit point functions (for [KRR17]) or exponentially secure KDM-secure secret key encryption with respect to arbitrary functions (for [CCRR18]).

Theorem 7.62. *Let $\Pi = (P, V)$ be a commit-challenge-response proof system for some language $L \in \mathbf{NP}$ with soundness error $\mu = \mu(n)$, where n denotes the length of a first message a. Moreover, let $|\Sigma| = |\Sigma(n)|$ be the number of possible challenges associated to a single commit message $a \in \{0,1\}^n$, let $N = \lambda|\Sigma|n$ (for arbitrarily related $n = \mathsf{poly}(\lambda)$), and suppose that*

- *$\mathcal{O}$ is a sub-exponential advantage secure indistinguishability obfuscator.*
- *$\mathcal{F} = \left\{\{F_{n,\lambda,s} : \{0,1\}^N \to \{0,1\}^{\lambda|\Sigma|\log|\Sigma|}\}_{s\in\{0,1\}^{\ell(n)}}\right\}_{n\in\mathbb{N}}$ is a family of $(\mathsf{poly}(N), 2^{-2N})$-secure puncturable PRFs. We will use the notation $F_s(\cdot)$ as shorthand for $F_{n,s}(\cdot)$.*
- *There* exists *a δ-secure injective OWF family $\mathcal{F}_{inj}$ for some $\delta = 2^{-N} \cdot (\frac{1}{\mu})^{\lambda|\Sigma|} \cdot \mathrm{negl}(N)$ taking inputs of length N.*

Then, if Π is instantiated using a public key encryption scheme to commit (where the public key is provided as a common reference string and commitment is encryption), then there is a polynomial $p : \mathbb{N} \to \mathbb{N}$ and a such that the hash family $\mathcal{H}$ defined by $H \leftarrow \mathcal{O}(P_s)$ instantiates the Fiat-Shamir heuristic[17] for a $\lambda|\Sigma|$-wise parallel repetition of Π, where P_s is a circuit (padded to size $p(n)$) that evaluates F_s.

Moreover, if Π is honest verifier zero-knowledge, then the new 1-message proof system Π' is also zero knowledge (with a programmable CRS).

Remark 7.63. *By Section 7.3.3, the same result holds if there exists a δ'-secure (not necessarily injective) OWF family for some $\delta'(N) = 2^{-N} \cdot 2^{\frac{N}{3}}\delta(\frac{N}{3})$.*

[17] in the common reference string model

Applying Theorem 7.62 to either the 3-colorability protocol of [GMW86] or the Hamiltonicity protocol of [FLS90] yields a construction of NIZK arguments (in the common reference string model). While NIZK proofs from IO and OWFs are already known by [BP15], this yields a construction of NIZK arguments through the Fiat-Shamir transform.

Proof of Theorem 7.62. Let x be any string *not* in the language L, and let crs be a random CRS for the commitment scheme used in Π. We would like to apply Theorem 7.57 to the single input-output relation

$$R = \left\{ R_\lambda := \left\{ (\mathbf{a}, \mathbf{e}) : \text{ there exists } \mathbf{z} \text{ such that } (\mathbf{a}, \mathbf{e}, \mathbf{z}) \text{ is an accepting transcript for } \Pi^{\lambda|\Sigma|} \right\} \right\},$$

which is a $\mu^{\lambda|\Sigma|}$-sparse relation for any $x \notin L$. Unfortunately, it is not clear that R satisfies the hypotheses of Theorem 7.57; namely, it is unclear whether R is efficiently samplable. This issue can be fixed with two modifications:

- We instantiate the commitment scheme using a public key encryption scheme, where the public key is provided as a common reference string.
- We replace the relation R_λ with a relaxed relation $\tilde{R}_{\lambda,\mathsf{sk}}$ that is in $\mathcal{R}_{1,0,\mu^{\lambda|\Sigma|}}$.

More specifically, the modified relation $\tilde{R}_{\lambda,\mathsf{sk}}$ is defined as follows:

$$\tilde{R}_{\lambda,\mathsf{sk}} = \left\{ (\mathbf{a}, \mathbf{e}) : \text{Check}\left(x, y^{(i)}_{T(e^{(i)})}, e^{(i)}\right) = 1 \text{ for all } i, \text{ where } \mathbf{y} = \mathsf{Dec}(\mathsf{sk}, \mathbf{a}) \right\}.$$

We first note that $\tilde{R}_{\lambda,\mathsf{sk}}$ is a strict relaxation (superset) of R_λ when the commitment scheme for Π is instantiated with a public key encryption scheme. This follows from (1) the definition of a commit-challenge-response protocol, and (2) the fact that given a first message $a^{(i)}$, the only possible valid decommitment to any block of $a^{(i)}$ is the corresponding block of $\mathsf{Dec}(\mathsf{sk}, a^{(i)})$.

Moreover, it is easy to see that $\tilde{R}_{\lambda,\mathsf{sk}}$ is efficiently (locally) samplable. The sampling algorithm is as follows: given $\mathbf{a} = a^{(1)}||\dots||a^{(\lambda|\Sigma|)}$ and sk, compute $\mathbf{y} =$

$\mathsf{Dec}(\mathsf{sk}, \mathbf{a})$. Then, for every $i \in [\lambda]$, do the following procedure: for every $e \in \Sigma$, run $\text{Check}(x, y^{(i)}_{T(e)}, e)$, and then sample $e^{(i)}$ uniformly at random from the set of e for which Check outputs 1. The sampling algorithm outputs $\mathbf{e} = e^{(1)}|| \ldots ||e^{(\lambda|\Sigma|)}$.

Since $\tilde{R}_{\lambda,\mathsf{sk}}$ is efficiently (locally) samplable and has sparsity $\mu^{\lambda|\Sigma|}$, we conclude that the hash family $\mathcal{H}$ is correlation intractable for $\tilde{R}_{\lambda,\mathsf{sk}}$ by Theorem 7.57. Moreover, since $\tilde{R}_{\lambda,\mathsf{sk}}$ is a relaxation of the relation R_λ, we conclude that it is hard for an efficient adversary $\mathcal{A}(H)$ to produce any message $\mathbf{a}$ such that $(\mathbf{a}, H(\mathbf{a}), \mathbf{z})$ is an accepting transcript for any possible $\mathbf{z}$. Thus, the Fiat-Shamir 1-message protocol is sound, as desired.

To show that the protocol is zero knowledge (if Π is honest verifier zero knowledge), we define the simulator Sim' for the 1-message protocol in terms of an honest-verifier simulator Sim for Π:

1. Sample a public key pk for the public key encryption scheme.

2. Run $\mathsf{Sim}(x, \mathsf{pk})$ independently $\lambda|\Sigma|$ times to obtain simulated transcripts $(\tilde{a}^{(i)}, \tilde{e}^{(i)}, \tilde{z}^{(i)})_{i \le \lambda|\Sigma|}$.

3. Letting $\tilde{\mathbf{a}} = (\tilde{a}^{(1)}, \ldots, \tilde{a}^{(\lambda|\Sigma|)})$, $\tilde{\mathbf{e}} = (\tilde{e}^{(1)}, \ldots, \tilde{e}^{(\lambda|\Sigma|)})$, and $\tilde{\mathbf{z}} = (\tilde{z}^{(1)}, \ldots, \tilde{z}^{(\lambda|\Sigma|)})$, compute the obfuscated program $\tilde{H} = \mathcal{O}(P_{s,\tilde{\mathbf{a}},\tilde{\mathbf{e}}})$, where

$$P_{s,\tilde{\mathbf{a}},\tilde{\mathbf{e}}}(x) = \begin{cases} \tilde{\mathbf{e}} & \text{if } x = \tilde{\mathbf{a}} \\ F_s(x) & \text{otherwise.} \end{cases}$$

4. Output $(\widetilde{\mathsf{CRS}}, \tilde{\pi}) = \Big((\mathsf{pk}, \tilde{H}), (\tilde{\mathbf{a}}, \tilde{\mathbf{z}})\Big)$.

The proof that Sim' samples from a distribution computationally indistinguishable from an honest proof follows by a hybrid argument: first, convert $(\tilde{\mathbf{a}}, \tilde{\mathbf{e}}, \tilde{\mathbf{z}})$ to a collection $(\mathbf{a}, \mathbf{e}, \mathbf{z})$ of $\lambda|\Sigma|$ honest Π-proofs by the security of Sim, and then convert the obfuscated program $\mathcal{O}(P_{s,\mathbf{a},\mathbf{e}})$ into an obfuscated program $\mathcal{O}(P_s)$ by obfuscation and puncturing security. □

7.7 A Proof of the Refined Asharov-Segev Bound

In this section, we prove Theorem 7.7 (a refined analysis of the Asharov-Segev impossibility result [AS15]). We now formally state Theorem 7.7.

Theorem 7.64. *There exists an oracle Γ' and an oracle $\Gamma = (\Gamma', \mathsf{CollFind}^{\Gamma'})$ such that no hash function built relative to Γ' is collision-resistant relative to Γ, and such that the following cryptographic primitives* can be built *relative to Γ' (and are secure relative to Γ):*

1. *$(2^{\frac{n}{15}}, 2^{-\frac{n}{40}})$-secure indistinguishability obfuscation*

2. *$(2^{\frac{n}{50}}, 2^{-\frac{n}{50}})$-secure one-way permutations.*

3. *A one-way permutation which is $(q(n), q(n)^c \cdot n \cdot 2^{-n})$-secure for every polynomial q (for some absolute constant c).*

As in [AS15], the oracle Γ is defined as follows.

Definition 7.65 (Asharov-Segev Oracle)**.** *The Asharov-Segev oracle $\Gamma = (\Gamma', \mathsf{CollFind}^{\Gamma'}) = \left(f, \mathcal{O},\ \mathsf{Eval}^{f,\mathcal{O}}, \mathsf{CollFind}^{f,\mathcal{O},\mathsf{Eval}}\right)$ consists of four parts:*

1. *A uniformly random permutation $f = f^{(n)} : \{0,1\}^n \to \{0,1\}^n$ for every input length n.*

2. *A uniformly random permutation $\mathcal{O} = \mathcal{O}^{(n)} : \{0,1\}^{2n} \to \{0,1\}^{2n}$ for every input length n.*

3. *The function $\mathsf{Eval}^{f,\mathcal{O}}$, on input $(z, x) \in \{0,1\}^* \times \{0,1\}^*$, finds the unique string $D||r$ such that $\mathcal{O}(D||r) = z$ and outputs $D^f(x)$. The combination of $\mathcal{O}$ and* Eval *will serve as our indistinguishability obfuscator.*

4. *A collision-finding oracle $\mathsf{CollFind}^{f,\mathcal{O},\mathsf{Eval}}$: on any input $C^{f,\mathcal{O},\mathsf{Eval}}$ (which is a circuit with $f, \mathcal{O}$, and* Eval*-gates),* CollFind *outputs a random $w \xleftarrow{\$} \{0,1\}^t$ (where t is the input length of C), as well as a uniformly random w' of the same input length subject to the condition that $C^{f,\mathcal{O},\mathsf{Eval}}(w) = C^{f,\mathcal{O},\mathsf{Eval}}(w')$.*

We refer the reader to [AS15] for details on Γ (in particular, on the specific implementation of $\mathsf{CollFind}$).

In [AS15], it is shown that CRHFs (implementable relative to Γ') do not exist relative to Γ (Claim 3.5 in [AS15]), $(2^{\frac{n}{15}}, 2^{-\frac{n}{40}})$-secure indistinguishability obfuscation exists relative to Γ (Theorem 3.8 in [AS15]), and $(2^{\frac{n}{50}}, 2^{-\frac{n}{50}})$-secure one way permutations exist relative to Γ (Theorem 3.20 in [AS15]). In particular, the one-way permutation they prove secure is f itself. We now strengthen their result to show that f is (nearly) 2^{-n}-secure.

Lemma 7.66. *Let $q(n)$ denote any polynomial function of n. Then, any adversary $\mathcal{A}^{\Gamma}$ which is given $y = f(x)$ (for $x \xleftarrow{\$} \{0,1\}^n$) and makes at most $q(n)$ queries to Γ (each of size at most $q(n)$) will output x with probability at most $q(n)^c \cdot n \cdot 2^{-n}$, for some absolute constant c. The probability here is taken over the choice of x as well as the choice of oracles $(f^{(n)}, \mathsf{CollFind})$ (but holds for any oracle $\mathcal{O}$).*

The rest of this section is devoted to establishing Lemma 7.66 with the help of [AS15]. The proof proceeds as follows.

Suppose that some adversary $\mathcal{A}^{\Gamma}$ is given $y = f(x)$ (for $x \xleftarrow{\$} \{0,1\}^n$) and outputs x with probability ϵ. Define $\textsc{Win}_{\mathcal{A}}$ to be the event that $f(\mathcal{A}^{\Gamma}(y)) = y$. Moreover, define the $\mathsf{CollHit}_{y,\mathcal{A}}$ to be the event that $\mathcal{A}$ makes *some call to the* $\mathsf{CollFind}$ *oracle* which outputs (w, w') with one of the following two properties:

1. Some f-gate in the circuit evalutation $C^{f,\mathcal{O},\mathsf{Eval}}(w)$ or $C^{f,\mathcal{O},\mathsf{Eval}}(w')$ has output y, OR
2. Some Eval-gate in $C^{f,\mathcal{O},\mathsf{Eval}}(w)$ or $C^{f,\mathcal{O},\mathsf{Eval}}(w')$ has input $(\hat{D}, a)$ such that $D^f(a)$ has an f-gate with output y, where D is the unique circuit such that $\mathcal{O}(D, r) = \hat{D}$ for some r.

Our refined analysis (as compared to [AS15]) is the following claim (and proof).

Claim 7.66.1. *Given $\mathcal{A}$ as above, there exists an algorithm $\mathcal{B}^{f,\mathcal{O},\mathsf{Eval},\mathsf{CollFind}}$ which makes at most $3q(n)^3$ queries to f, $q(n)$ queries to* Eval*, and $q(n)$ queries to* $\mathsf{CollFind}$*,* *such that*

$$\Pr[\textit{\textsc{Win}}_{\mathcal{B}} \wedge \overline{\mathsf{CollHit}}_{y,\mathcal{B}}] \geq \frac{\epsilon}{6}.$$

Proof. We may assume that

$$\Pr[\textsc{Win}_{\mathcal{A}} \wedge \mathsf{CollHit}_{y,\mathcal{A}}] \geq \frac{\epsilon}{2},$$

because otherwise we may just set $\mathcal{B} = \mathcal{A}$. In the remaining case, we define $\mathcal{B}$ as follows: $\mathcal{B}^{f,\mathcal{O},\mathsf{Eval},\mathsf{CollFind}}(y)$ executes $\mathcal{A}^{f,\mathcal{O},\mathsf{Eval},\mathsf{CollFind}}(y)$, except that whenever $\mathcal{A}$ would make a query C to $\mathsf{CollFind}$, it first samples a random $z \leftarrow \{0,1\}^t$ (where t is the input length of the circuit C), explicitly evaluates $C^{f,\mathcal{O}}(z)$ *without invoking* Eval[18], and checks if this evaluation has any f-gate with output y. If so, $\mathcal{B}$ returns the input to this f-gate and halts; otherwise, $\mathcal{B}$ continues the execution of $\mathcal{A}$.

It was already noted in [AS15] that $\mathcal{B}$ makes at most $3q(n)^3$ queries to f, and at most $q(n)$ queries to Eval and $\mathsf{CollFind}$, respectively. To prove the desired inequality, we define $\mathsf{Guess}_{y,\mathcal{B}}$ to be the event that $\mathcal{B}$ successfully inverts y in one of its z-experiments as described above. Then, we see that

$$\Pr[\textsc{Win}_{\mathcal{B}} \wedge (\mathsf{Guess}_{y,\mathcal{B}} \vee \mathsf{CollHit}_{y,\mathcal{B}})] \geq \Pr[\textsc{Win}_{\mathcal{A}} \wedge \mathsf{CollHit}_{y,\mathcal{A}}] \geq \frac{\epsilon}{2}.$$

This inequality follows by considering a third algorithm $\mathcal{C}$ which acts as $\mathcal{B}$ does but *does not halt* after any z-experiment ($\mathcal{C}$ instead entirely ignores the outcome of this experiment); it is clear that

$$\Pr[\textsc{Win}_{\mathcal{B}} \wedge (\mathsf{Guess}_{y,\mathcal{B}} \vee \mathsf{CollHit}_{y,\mathcal{B}})] \geq \Pr[\textsc{Win}_{\mathcal{C}} \wedge \mathsf{CollHit}_{y,\mathcal{C}}] = \Pr[\textsc{Win}_{\mathcal{A}} \wedge \mathsf{CollHit}_{y,\mathcal{A}}].$$

Next, we show that

$$\Pr[\textsc{Win}_{\mathcal{B}} \wedge \mathsf{Guess}_{y,\mathcal{B}} \wedge \overline{\mathsf{CollHit}}_{y,\mathcal{B}}] \geq \frac{1}{2}\Pr[\textsc{Win}_{\mathcal{B}} \wedge \mathsf{CollHit}_{y,\mathcal{B}}].$$

[18] In other words, for every query $(\hat{D}, a)$ to Eval, $\mathcal{B}$ will make exponentially many calls to $\mathcal{O}$ to brute-force recover D from $\hat{D}$, and then evaluate $D^f(a)$.

To see this, we write

$$\Pr[\textsc{Win}_{\mathcal{B}} \wedge \mathsf{Guess}_{y,\mathcal{B}} \wedge \overline{\mathsf{CollHit}}_{y,\mathcal{B}}] = \sum_{i=1}^{q} \Pr[\mathsf{Guess}_i],$$

where Guess_i is the event that $\mathcal{B}$ does *not* invert y in the first $i-1$ z-experiments it runs, does *not* invert y with one of the first $i-1$ $\mathsf{CollFind}$ queries, but *does* invert y in the ith z-experiment. Similarly, we write

$$\Pr[\textsc{Win}_{\mathcal{B}} \wedge \mathsf{CollHit}_{y,\mathcal{B}}] \leq \sum_{i=1}^{q} [\mathsf{CollHit}_i],$$

where $\mathsf{CollHit}_i$ is the event that $\mathcal{B}$ does *not* invert y in the first $i-1$ z-experiments it runs, does *not* invert y with one of the first $i-1$ $\mathsf{CollFind}$ queries, but *does* invert y in its ith $\mathsf{CollFind}$ query. Our claim now follows from the inqualities

$$\Pr[\mathsf{Guess}_i] \geq \frac{1}{2} \Pr[\mathsf{CollFind}_i],$$

which holds because given that no inversion has occurred in the first $i-1$ z-experiments and $\mathsf{CollFind}$ queries, the probability that the ith $\mathsf{CollFind}$ query produces (w, w') leading to a y-inversion is at most twice the probability that w (the first input) leads to a y-inversion, which is identical to the probability that the ith z-experiment leads to a y-inversion (because z and w are both just uniformly random inputs to C_i, the ith $\mathsf{CollFind}$ query).

Finally, we conclude the desired result by the calculation

$$\begin{aligned}
\Pr[\textsc{Win}_{\mathcal{B}} \wedge \overline{\mathsf{CollHit}}_{y,\mathcal{B}}] &\geq \Pr[\textsc{Win}_{\mathcal{B}} \wedge \mathsf{Guess}_{y,\mathcal{B}} \wedge \overline{\mathsf{CollHit}}_{y,\mathcal{B}}] \\
&\geq \frac{1}{3} \Pr[\textsc{Win}_{\mathcal{B}} \wedge (\mathsf{Guess}_{y,\mathcal{B}} \vee \mathsf{CollHit}_{y,\mathcal{B}})] \\
&\geq \frac{1}{3} \Pr[\textsc{Win}_{\mathcal{A}} \wedge \mathsf{CollHit}_{y,\mathcal{A}}] \\
&\geq \frac{\epsilon}{6}.
\end{aligned}$$

□

To conclude Theorem 7.64, we combine Claim 7.66.1 with the following additional claim from [AS15] (minimally modified).

Claim 7.66.2 ([AS15], Claim 3.27)**.** *If any algorithm $\mathcal{B}$ makes at most Q queries to f,* Eval*, and* $\mathsf{CollFind}$ *(each), then*

$$\Pr[\textit{W}\textsc{\textit{in}}_{\mathcal{B}} \wedge \overline{\mathsf{CollHit}}_{y,\mathcal{B}}] \leq \delta + 2^{\frac{-\delta 2^n}{3Q(n)^3}}$$

for every $\delta > 0$.

In particular, setting $\delta = 3n \cdot Q(n)^3 2^{-n}$, we see that

$$\Pr[\textsc{Win}_{\mathcal{B}} \wedge \overline{\mathsf{CollHit}}_{y,\mathcal{B}}] \leq (3n \cdot Q(n)^3 + 1)2^{-n}$$

for any such $\mathcal{B}$. Using the $\mathcal{B}$ we produced from $\mathcal{A}$ in Claim 7.66.1, we conclude that

$$\frac{\epsilon}{6} \leq \Pr[\textsc{Win}_{\mathcal{B}} \wedge \overline{\mathsf{CollHit}}_{y,\mathcal{B}}] \leq (81n \cdot q(n)^3 + 1)2^{-n},$$

yielding the desired bound on $\epsilon = \Pr[\textsc{Win}_{\mathcal{A}}]$, and hence Theorem 7.64.

Chapter 8

Correlation-Intractable Hash Functions via Shift Hiding

8.1 Introduction

The random oracle model [BR93] is a powerful but controversial paradigm in cryptography in which the proof of security of a cryptographic scheme assumes that a certain publicly computable function H that is used in the scheme behaves like a random function to the adversary. The random oracle model is hugely influential in designing concretely efficient cryptosystems, but is inherently problematic theoretically: how could a *public*, and therefore completely predictable, function behave in all aspects like a random function? Indeed, Canetti, Goldreich and Halevi [CGH98] demonstrated cryptographic schemes that one could prove secure in the random oracle model, but which are insecure no matter how one tries to instantiate the oracle with a concrete function (or even a function chosen at random from an exponential-size family). Nevertheless, this negative result and the notions introduced therein led to a long line of research that asked *what concrete properties* of a random oracle are instantiable in the standard model (see, e.g., [CMR98] for an early work in this direction), and opened the door to groundbreaking positive results two decades later [CCR16, KRR17, CCRR18, HL18, CCH+19, PS19].

The key notion introduced in [CGH98] is that of correlation intractability (CI), which captures a general and powerful form of cryptographic hardness for a hash family $\mathcal{H}$. For any binary relation $R(x, y)$, a hash family $\mathcal{H}$ is correlation-intractable for R if it is computationally hard (given a hash function $h \leftarrow \mathcal{H}$) to find an input x such that $R(x, h(x))$ is true. For this definition to make sense, we require that the relation R is sparse: for any x, all but a negligible fraction of y do not satisfy the relation with x.

For decades, there was little progress on building correlation-intractable hash functions in the standard model outside of a few extremely simple cases (such as one-way functions). However, there has been much recent work [CCR16, KRR17, CCRR18, HL18, CCH+19, PS19, BKM20, LV20a] on instantiating restricted but expressive variants of CI. Namely, these works made the following simplifications:

- Starting with [CCR16, HL18], additional *efficiency* requirements were placed on the relation R. For example, one can require that $R(x, y)$ is decidable in (bounded) polynomial time.

- Starting with [CCH+19], the relation R was further specialized to represent an *efficiently computable function* f. A hash family $\mathcal{H}$ is CI for f if it is hard, given h, to find an input x such that $h(x) = f(x)$.

While these restrictions may seem extreme, these limited forms of CI remain expressive and powerful. In particular, even CI for efficiently computable functions has implications for the instantiability of the Fiat-Shamir transform [FS87] in the standard model [DNRS99, BLV03, CCR16] for constant-round public-coin interactive proof systems. Most notably, [CCH+19, PS19] construct hash families $\mathcal{H}$ that are CI for efficiently computable functions under standard cryptographic assumptions related to the learning with errors (LWE) problem, and use these hash families to build the first lattice-based non-interactive zero-knowledge (NIZK) proof systems for NP.

Let us recall the [CCH+19, PS19] constructions at a high level. [CCH+19] gives a *generic* construction using fully homomorphic encryption (FHE) [Gen09, BV11]. The construction is simple: a hash function $h \leftarrow \mathcal{H}$ is parameterized by a FHE ciphertext

$\mathsf{Enc}(g)$ for some (dummy) function g. To evaluate $h(x)$, simply homomorphically evaluate g on x to obtain some ciphertext of the form $\mathsf{Enc}(g(x))$. One can show that this hash family is CI for a function f if the FHE scheme is *circular secure*: since g is computationally hidden, we can replace it in the security proof with a function $g^*(x) = \mathsf{Dec}_{\mathsf{sk}}(f(x)) + 1$ specifically designed to avoid $f(x)$ at the ciphertext level.

While this construction is both simple and generic, it has the significant drawback that it relies on the circular security (rather than semantic security) of the FHE, and therefore cannot be proven secure under the plain LWE assumption. Peikert and Shiehian [PS19] then gave an ingenious construction of CI based on plain LWE. Their construction uses the algebra of the [GSW13] FHE scheme to give a special-purpose variant of the [CCH+19] approach that avoids reliance on circular security. However, this requires making a number of changes to the hash function: at a high level, they "downgrade" plain LWE-based GSW ciphertexts after evaluation to Regev "ciphertexts" (where the plaintext space is $\mathbb{Z}_q$ and decryption correctness is only approximate) with circular dependencies. This results in a LWE-based CI hash family, but loses the conceptual simplicity of the [CCH+19] construction.

8.1.1 Our Results and Techniques

Our main result is a new framework for constructing CI hash functions using a cryptographic primitive called *shift-hiding shiftable functions* (SHSFs) [PS18], a twist on private constrained pseudorandom functions [BW13, BGI14, KPTZ13]. A SHSF family is a function family $\{F_{\mathsf{msk}}\}$ that additionally supports the ability to *delegate* a constrained key sk_f that enables computation of the map $x \mapsto F_{\mathsf{msk}}(x) + f(x)$, without revealing the "shift function" f. Shift-hiding shiftable functions were originally introduced for the purpose of constructing private constrained PRFs, but have since found several other applications [PS20, DVW20].

In a nutshell, we show that SHSFs are intimately tied to correlation intractability via an extremely short proof. We further develop this framework in three directions.

1. We obtain a conceptually simple construction of CI for functions based on LWE.

This construction can replace the FHE-based approach of [CCH+19, PS19] and shows that the prior function family of [PS18] (constructed for an entirely different purpose) was *already* a good CI hash family.

2. We show that our construction transparently generalizes to new variants of *multi-input* CI, which is currently poorly understood.

3. We give additional instantiations of our framework (which are new, in both the single- and multi-input settings) using indistinguishability obfuscation and other standard assumptions.

Moreover, we believe that our framework and new approach to constructing CI hash functions may be useful for future progress on and understanding of this primitive.

Lifting CI. We begin with a description of (1). Our main technique is a *lifting theorem* (Theorem 8.23) that allows us to construct CI hash functions for complex relations starting from CI hash functions for simpler relations. In the single-input setting, it states that any SHSF family (for a function class $\mathcal{F}$) satisfying a *very weak* form of correlation intractability is essentially already a CI hash family for $\mathcal{F}$.

Theorem 8.1 (Informal)**.** *Suppose that* $\mathsf{SHSF} = \{F_{\mathsf{msk}}\}$ *is a family of SHSFs for a function class* $\mathcal{F}$*, and suppose that* F_{msk} *satisfies either of the following two* one-wayness *properties:*

- *Given* msk*, it is hard to find an element in* $F_{\mathsf{msk}}^{-1}(0)$*,* or
- *Given* msk *and a uniformly random target* r*, it is hard to find an element in* $F_{\mathsf{msk}}^{-1}(r)$*.*

Then, the shifted evaluation algorithm of SHSF *describes a hash family* $\mathcal{H}$ *that is correlation-intractable for all functions* $f \in \mathcal{F}$*.*

The CI hash function is extremely simple to describe. Hash keys are shifted keys $\mathsf{sk}_{\mathcal{Z}}$ for the all-zero function $\mathcal{Z}$, and hash function evaluation is simply the shifted evaluation using $\mathsf{sk}_{\mathcal{Z}}$ which computes exactly the function F_{msk}. (Philosophically, the CI hash family constructed in this theorem is a form of "obfuscated PRF evaluation" although shift-hiding functions are decidedly more complex to construct than PRFs.) The proof of Theorem 8.1 is also simple.

Proof Sketch. If an adversary $\mathcal{A}$, given a hash key $\mathsf{sk}_{\mathcal{Z}}$, finds an input x such that

$$\mathsf{Hash}(x) := F_{\mathsf{sk}_{\mathcal{Z}}}(x) = f(x) \ ,$$

then by the shift-hiding property of SHSF, $\mathcal{A}$ also produces such an x when given sk_f instead of $\mathsf{sk}_{\mathcal{Z}}$. In that case, $\mathcal{A}$ solves the equation

$$f(x) = F_{\mathsf{sk}_f}(x) = F_{\mathsf{msk}}(x) + f(x),$$

which is equivalent to the equation $F_{\mathsf{msk}}(x) = 0$. This yields a 0-inversion attack on F_{msk}. The "random target" version of the theorem holds by the same argument, using a shifted key sk_{f_r} for the function $f_r(x) = f(x) - r$. □

We note that Theorem 8.1 could be proved under a weaker one-wayness assumption, namely, that *it is hard to find an input x such that $F_{\mathsf{msk}}(x) = 0$, given a shifted key sk_f for any pre-specified f*" (as opposed to being given msk in the clear). However, we phrase Theorem 8.1 under the assumption that F_{msk} is one-way (given msk in the clear) because this is a clean, f-independent security property, which also makes it more amenable to instantiation/proof. In our constructions below, we prove the stronger one-wayness property of F_{msk}.

Instantiation from LWE. Given Theorem 8.1, it remains to construct an SHSF family satisfying this one-wayness property. We show that a variant of the Peikert-Shiehian SHSF [PS18] satisfies this.

Theorem 8.2 (Informal, see Theorem 8.24)**.** *Assuming the hardness of standard* *lattice problems (LWE and 1-dimensional SIS variants), the [PS18] SHSF*[1] *is one-way.*

We now sketch our proof assuming some knowledge of LWE-based cryptography.

Proof Sketch. In the Peikert-Shiehian SHSF construction, $\mathsf{msk} = \mathbf{s} \in \mathbb{Z}_q^n$ is an LWE secret, and

$$F_{\mathsf{msk}}(x) = \lfloor \mathbf{s}\mathbf{A}_x + \mathbf{u} \cdot \mathbf{G}^{-1}(\mathbf{A}_x) \rceil_p \in \mathbb{Z}_p^{\mu}$$

where $\mathbf{G} \in \mathbb{Z}_q^{n\times m}$ is the gadget matrix, $\mathbf{u} \in \mathbb{Z}_q^m$ is a uniformly random row vector, $\mathbf{A}_x \in \mathbb{Z}_q^{n\times\mu}$ is a matrix constructed out of (uniformly random) matrices $\mathbf{A}_1, \dots, \mathbf{A}_\ell$ using the gadget homomorphisms from [BGG+14], and $\lfloor\cdot\rceil_p$ denotes the rounding operation that (roughly speaking) keeps the top $\log p$ bits of the argument and discards the rest. By [PS18], this family is shift-hiding under the LWE assumption and (computationally) correct under the 1D-SIS assumption (Definition 8.21).

If the adversary finds an x such that $F_{\mathsf{msk}}(x) = 0$, there are two cases; the first case is when $\mathbf{G}^{-1}(\mathbf{A}_x)$ is non-zero. This gives an approximate subset sum solution for the instance $\mathbf{s}\mathbf{G} + \mathbf{u}$, that is,

$$(\mathbf{s}\mathbf{G} + \mathbf{u})\mathbf{G}^{-1}(\mathbf{A}_x) \in q\mathbb{Z}^{\mu} + [-\frac{q}{p}, \frac{q}{p}]^{\mu}.$$

This violates (on whichever column of $\mathbf{G}^{-1}(\mathbf{A}_x)$ is nonzero) a natural one-dimensional variant of SIS (Definition 8.19) that we show is as hard as worst-case lattice problems provided that p is large enough[2] (see Section 8.2.3).

The second case is when the adversary finds an x such that $\mathbf{G}^{-1}(\mathbf{A}_x) = 0$, which implies that $\mathbf{A}_x = 0$. We show that the adversary cannot make this happen without violating SIS (again!) Roughly speaking, we use the fact that if we *program* the matrices $\mathbf{A}_i = \mathbf{A}\mathbf{R}_i + h_i\mathbf{G}$ where $\mathbf{R}_i$ are matrices with small entries and h is the description of a constant function with image $y \neq 0 \in \mathbb{Z}_q^{\mu}$, the following equation

[1] Compared to [PS18], (1) our construction is slightly modified for ease of proof, and (2) particular parameter settings are required.

[2] Some care must be taken to set parameters so that the SHSF security reductions still hold for this choice of p.

holds for each column $\mathbf{a}_x^{(j)}$ of $\mathbf{A}_x$ due to the gadget homomorphisms of Boneh et al. [BGG+14]:

$$\mathbf{a}_x^{(j)} = \mathbf{A}\mathbf{r}_x^{(j)} + y_j \mathbf{u}_1$$

(where $\mathbf{u}_1$ is the first standard basis vector) for some $\mathbf{r}_x^{(j)}$ that is a function of $\mathbf{R}_1, \ldots, \mathbf{R}_\ell$. We know by assumption that $\mathbf{A}_x = 0$. Since $y \neq 0$, this means that the adversary found a valid solution $\mathbf{R}_x = \begin{bmatrix} \mathbf{r}_x^{(1)} & \ldots & \mathbf{r}_x^{(\mu)} \end{bmatrix}$ to the (inhomogenous) SIS problem $\mathbf{A}\mathbf{R}_x = -\mathbf{u}_1 y^\top \in \mathbb{Z}_q^{n \times \mu}$, which is hard assuming that worst-case lattice problems are hard. This finishes the proof of one-wayness. □

Combining Theorem 8.2 with Theorem 8.1, we already recover a similar result to [PS19]. That is, assuming the hardness of standard lattice problems, there exists a hash family that is correlation-intractable for all bounded-size functions. By appealing to [CCH+19], this also gives a lattice-based NIZK argument system for NP. However, our approach leverages this new, conceptually simple connection to SHSFs and shows that [PS18] were "most of the way" to LWE-based CI. Besides the extremely simple bootstrapping theorem, the missing piece was whether a natural PRF construction [PS18] satisfies a one-wayness property given msk in the clear. A similar question was previously studied for the GGM PRF family [CK16], but does not appear to have been addressed for other concrete PRF families.

Next, we describe how our techniques extend to give new feasibility results in two different directions:

- They immediately generalize to setting of *multi-input* CI, and
- They allow for new generic instantiations based on indistinguishability obfuscation.

We remark that constructing (single- or multi-input) CI hash functions even assuming indistinguishability obfuscation is far from straightforward. Indeed, the initial works [CCR16, KRR17, HL18] in this line all made non-standard assumptions *in addition to iO*. Non-standard assumptions were required until the work of [CCH+19] which constructed single-input CI hash functions under circular-secure LWE. However, they only managed to do this for a tiny subset of relations that [CCR16,KRR17] achieved. In particular, replicating the results of [KRR17] or even [CCR16] assuming *only iO* (plus standard assumptions) is a challenging open problem.

8.1.2 Applications: Multi-Input CI from LWE and CI from iO

So far, we have only discussed *single-input* CI; that is, we considered CI for relations with a single input x and single corresponding output y. However, there is a natural generalization of CI to relations with many input-output pairs: a hash family $\mathcal{H}$ is defined to be CI for a relation $R(x_1, \ldots, x_t, y_1, \ldots, y_t)$ if it is computationally hard (given a hash function $h \leftarrow \mathcal{H}$) to find inputs $x_1, \ldots, x_t$ such that $(x_1, \ldots, x_t, h(x_1), \ldots, h(x_t)) \in R$. In contrast to the single-input case, *multi-input* correlation intractability (for any $t \geq 2$) is a far less well-understood primitive. Perhaps the simplest nontrivial example of multi-input CI is for the relation R where $R(x_1, x_2, y_1, y_2) = 1$ if and only if $y_1 = y_2$ but $x_1 \neq x_2$. A CI hash family for R is precisely a collision-resistant hash family. However, most multi-input relations do not correspond to security notions that are simple-to-understand or previously studied. CI for more general multi-input relations also has interesting applications, including:

1. As a useful tool for the *untrusted setup* of public parameters [CCR16, Zha16]: Multi-input CI hash functions allow n parties $P_1, \ldots, P_n$ with inputs $x_1, \ldots, x_n$ to compute public outputs $y_i = H(x_i)$ that can be used to generate public parameters for a multi-party protocol. Correlation intractability of H is necessary to ensure that a "bad CRS" is not accidentally (or maliciously) agreed on.

2. As a hash function in proof-of-work protocols [CCR16,CCRR18]: In the bitcoin protocol [Nak08], a miner succeeds in adding a block to the blockchain when she finds an x such that $y = H(x||B_i)$ starts with a specified number of zeroes (here, B_i is the i-th block and once found, y is placed in the next block B_{i+1}). A very desirable property in this setting is that a single miner (or collection

of colluding miners) cannot find *multiple consecutive blocks* with significantly less effort than finding them sequentially. This property can be formalized as a quantitatively precise[3] variant of multi-input CI. For example, in the case of two consecutive blocks, simplifying the setting a little, we require a 2-input CI for the relation R where $R(x_1, x_2, y_1, y_2) = 1$ iff y_1 and y_2 start with a pre-specified number ℓ of zeroes, and y_1 is a suffix of x_2.

Unfortunately, multi-input CI has so far proved hard to achieve. In particular, the constructions of [CCR16, KRR17, CCRR18, CCH+19, PS19, BKM20] are only known to achieve single-input CI. Holmgren and Lombardi [HL18] do achieve multi-input CI for a large class of relations that they call *locally sampleable* relations. However, they require both an indistinguishability obfuscation (iO) scheme [BGI+01] as well as an "optimally-secure" one-way product function [HL18]. While iO can now be achieved under relatively standard assumptions [JLS21, GP21, BDGM20b, WW21], the latter is a very strong "brute force is optimal"-type assumption. Zhandry [Zha16] constructed a hash family satisfying a very special form of multi-input CI called "output intractability". Output intractability is a form of CI for relations $R(x_1, \ldots, x_t, y_1, \ldots, y_t)$ that depend only on the y_i, which captures some variants of application (1) above. On the plus side, the construction is based on the exponential hardness of the Diffie-Hellman problem.[4] To summarize, multi-input CI is either known for a small class of relations under standard assumptions, or for a larger class of relations under very strong assumptions. We refer the reader to Section 8.1.3 for more details and further comparisons.

Multi-Input CI via Shift-Hiding. One consequence of our shift-hiding technique is a collection of feasibility results for multi-input correlation intractability based on standard assumptions. We obtain two flavors of results: constructions from standard

[3] As noted in [CCR16], CI following the $(\mathsf{poly}, \mathsf{negl})$ security definition framework is insufficient for this application. Instead, these protocols desire a concrete "moderately small" probability of breaking CI and a tight gap between honest and adversarial parties' probabilities of doing so in a fixed runtime. We do not attempt to address this subtlety in this work.

[4] Moreover, given an inverse-subexponential lower bound on the sparsity of the relation, Zhandry's construction is secure under (the more standard) sub-exponential DDH.

(lattice) assumptions, and constructions from indistinguishability obfuscation.

Our results are obtained via a generalization of our lifting theorem (Theorem 8.1) to multi-input relations. This gives us three new constructions of multi-CI hash functions under different assumptions:

- Our first construction considers the shifted linear relation

  $$\mathcal{R}_{\mathsf{lin}} = \{(x_1, \ldots, x_t, y_1, \ldots, y_t) : \sum w_i y_i = \sum w_i f(x_i) \pmod{p}\}$$

  where p is some large integer (roughly 2^λ), w_i are small weights and f is an arbitrary polynomial-time computable function. We construct a multi-input CI hash function for $\mathcal{R}_{\mathsf{lin}}$ under the same lattice assumptions as in the single-input case (all approximation ratios are larger by a factor of t).

- Our second and third constructions consider the shifted general relation

  $$\mathcal{R} = \{(x_1, \ldots, x_t, y_1, \ldots, y_t) : \mathcal{R}_0(y_1 - f(x_1), \ldots, y_i - f(x_i)) = 1\}$$

  where $\mathcal{R}_0$ is any polynomial-time decidable relation. In particular, our second construction achieves a multi-input CI hash function for $\mathcal{R}$ under subexponential iO, subexponential OWFs, and (sufficiently) lossy functions.

Our Generalized Lifting Theorem. Given any output-only relation $\mathcal{R}_0$, we say that a hash family $\mathcal{H}$ is $\mathcal{R}_0$-output intractable if it is hard (given h) to find distinct[5] inputs $x_1, \ldots, x_t$ such that $(y_1, \ldots, y_t) \in \mathcal{R}_0$ for $y_i = h(x_i)$. Output intractability as a standalone property (like collision-resistance) is known to be instantiable based on standard cryptographic assumptions (e.g., lossy functions [PW08]) as we discuss in Section 8.1.3. Our generalization of Theorem 8.1 states that *SHSFs that are output-intractable* lead to interesting new CI constructions.

[5] For the relation $\sum_i w_i y_i = 0$ implicitly described above, it is enough to assume that the inputs x_i are not all equal for the relation to be sparse. We elaborate on this weakening of output intractability as compared to [Zha16, HL18] in Section 8.2.

Theorem 8.3 (Also see Theorem 8.23)**.** *Suppose that* SHSF *is a shift-hiding shiftable function family. Assume that it is hard, given* msk*, to find distinct* $x_1, \ldots, x_t$ *such that* $\mathcal{R}_0(y_1, \ldots, y_t) = 1$ *where* $y_i = F_{\mathsf{msk}}(x_i)$ *and* $\mathcal{R}_0$ *is some polynomial-time computable relation. Then, there is a CI hash family for the shifted output relation*

$$\mathcal{R} = \{(x_1, \ldots, x_t, y_1, \ldots, y_t) : \mathcal{R}_0(y_1 - f(x_1), \ldots, y_i - f(x_i)) = 1\}$$

The proof of Theorem 8.3 follows from that of the single-input CI case *mutatis mutandis*. Thus, all that remains is to construct SHSFs that are *output-intractable*. We show three constructions.

Instantiation from LWE. To obtain a form of multi-input CI from LWE, we combine Theorem 8.3 with a generalization of Theorem 8.2:

Theorem 8.4. *Under standard lattice assumptions, there exists a SHSF family* SHSF *satisfying the following form of correlation intractability: for every nonzero vector* $w \in \{-1, 0, 1\}^t$*, it is hard (given* msk*) to find* t *distinct inputs* $x_1, \ldots, x_t$ *such that*

$$\sum_i w_i \cdot F_{\mathsf{msk}}(x_i) = 0,$$

where the sum is computed modulo some (large enough) integer p*.*

Our modification of the Peikert-Shiehian [PS18] construction satisfies this more general form of output intractability (for small linear equations), although the proof (in “Case 2” above) is more complicated (see Section 8.4.5). Note that this is a strict generalization of both single-input CI for functions (where $t = 1, w = 1$) and collision-resistance (where $t = 2, w = (-1, 1)$ and f is the constant function). Previously, this form of correlation intractability was only known assuming iO and (extremely hard) one-way product functions [HL18].

Instantiation from IO + lossiness. Our second construction achieves correlation intractability for shifted $\mathcal{R}_0$-output relations for a large class of $\mathcal{R}_0$ simultaneously (as opposed to linear $\mathcal{R}_0$ as in the LWE case above). It can be thought of as a (non-black-box) combination of our approach with a construction due to Zhandry [Zha16] of output-intractable hash functions.

Theorem 8.5. *Assume the existence of subexponential iO, subexponential OWFs, and lossy functions with input domain $\{0,1\}^n$ with a range of size $\leq 2^\ell$ in lossy mode. Then, there exists a hash family $\mathcal{H}$ that is CI for all (efficiently decidable) shifted t-ary output relations with sparsity at most $2^{-t\ell}$.*

As a corollary, we conclude that additionally assuming the existence of *extremely lossy functions* [Zha16], there is a hash family $\mathcal{H}$ that is CI for all (efficiently decidable) shifted t-ary output relations with sparsity $2^{-\omega(t)}$. As another corollary, we note that by combining Theorem 8.5 with [CCH+19], we obtain a construction of dual-mode NIZKs for NP based on iO, (injective) lossy functions, and lossy encryption. This closely matches the assumptions used in the work [HU19] but with a simpler construction. The corollary follows because the hash family from Theorem 8.5 satisfies "somewhere statistical correlation intractability."

A Separation between Single-Input and Multi-Input CI. Finally, we show that single-input and multi-input CI hash functions are fundamentally different primitives by demonstrating a separation between them. This follows from our third new CI instantiation, which is interesting even in the single-input setting.

Theorem 8.6. *Assume the existence of subexponentially secure indistinguishability obfuscation, subexponentially secure one-way functions, and a hash family $\mathcal{H}$ such that $\mathcal{H}$ is $\mathcal{R}_0$-output intractable, and for a random input X, $h_k(X)$ is 2^{-n}-indistinguishable from uniform (even given k). Then, there exists a hash family that is CI for shifted $\mathcal{R}_0$-relations.*

This theorem says that assuming subexponential iO and one-way functions, shifted-CI for $\mathcal{R}_0$ can be constructed (semi-)generically from output intractability for $\mathcal{R}_0$. Theorem 8.6 is proved by combining Theorem 8.3 with a construction of an $\mathcal{R}_0$-

output intractable SHSF using iO, puncturable PRFs, and an output-intractable hash function satisfying the above statistical requirement.

We note that as a corollary to Theorem 8.6, we obtain a construction of single-input CI for all efficient functions from iO and one-way permutations.[6]

Corollary 8.7. *If subexponential iO, subexponential OWFs, and (polynomially-secure) OWPs exist, then there exists a hash family that is CI for all efficient functions, that is, relations $\mathcal{R}(x, y)$ which is true iff $y = f(x)$.*

Corollary 8.7 follows from Theorem 8.6 by setting the output-intractable hash function $\mathcal{H}$ to be $h_k(x) := f(x) + k$, where f is a one-way permutation[7] and k is a uniformly random key. This construction is notable in that it separates *single-input* correlation intractability (theoretically) from *two-input* correlation intractability: due to an impossibility result of Asharov-Segev [AS15], it is known that there is no (black-box) construction of CRHFs from iO and one-way permutations (even with exponential security). A similar separation was shown in [HL18], but the "positive result" required assuming *optimally hard* one-way functions along with iO to obtain CI for all efficient functions (and more). In contrast, our construction is based on assumptions in the quantitatively standard regime.

8.1.3 Additional Related Work Discussion

Multi-Input Correlation Intractability We summarize what was previously known regarding multi-input correlation intractability:

- For subexponentially sparse output relations $\mathcal{R}_0$, output intractability for $\mathcal{R}_0$ can be constructed based on lossy functions (following [Zha16], but relying on less extreme forms of lossiness). Based on "extremely lossy functions",

[6] As is common [GR13], one must be careful about which definitions of "one-way permutation" suffice for this result. In our proof (which suffices for the separation), we assume that the one-way permutation has domain $\{0, 1\}^n$. It turns out that the proof can be made to work for discrete log-based one-way permutations, but does *not* appear to work for the (trapdoor) permutations constructed based on iO [BPW16].

[7] It suffices for f to be a OWF whose output distribution is close to uniform, e.g., a surjective regular OWF.

Zhandry [Zha16] constructs a hash family that is CI for all sparse (efficiently decidable) output relations.[8]

- Similarly to Zhandry [Zha16], the construction $x \mapsto p(H_k(x))$ (where H_k is a sufficiently shrinking collision-resistant hash function and p is sampled from a t-wise independent hash family) also yields output intractability for subexponentially sparse (and efficiently decidable) output relations.

- Holmgren and Lombardi [HL18] construct output-intractable hash functions for all sparse (even inefficient) R based on "one-way product functions" (OWPFs), OWFs satisfying a quantitiatively extreme assumption about the hardness of inverting many one-way function challenges in parallel. OWPFs (in different parameter regimes) are existentially incomparable to lossy functions and CHRFs. Under sufficiently strong assumptions, these hash families achieve quantitiatively better security than is possible for the previous two constructionss.

- Holmgren and Lombardi [HL18] also construct correlation-intractable hash families for relations $R(\mathbf{x}, \mathbf{y})$ that include all shifted output relations. However, they rely on both indistinguishability obfuscation and OWPFs (as above).

Comparison with Peikert-Shiehian [PS19]. [PS19] constructs single-input CI based on the LWE (or SIS) assumption. Their construction improves upon the construction of [CCH$^+$19] based on circular-secure FHE: by making use of special properties of the [GSW13] (and related) FHE schemes, they can remove the need for a circular ciphertext $\mathsf{Enc}(\mathsf{sk}, \mathsf{sk})$ in a specific GSW-based construction. By comparison, we show that any SHSF that is one-way is also CI for bounded functions, and that (essentially) the [PS18] SHSF is one-way. It does not seem easy to abstract out a simple, generic property of the [PS19] hash function that implies multi-input correlation intractability.

Given our generalization to multi-input CI, it is also reasonable to ask whether the [PS19] hash function also satisfies a form of multi-input CI. In fact, it appears likely

[8] This is a special case of Zhandry's actual result; we refer the reader to [Zha16] for more details.

that it satisfies CI for shifted-sum relations (just like our construction). However, a proof of this fact requires some of our analysis in the security proof of our multi-input CI construction (Theorem 8.4).

Comparison with Brakerski-Koppula-Mour [BKM20]. We also note that our construction shares some conceptual similarity to the recent CI construction of [BKM20]. We highlight the similarity here:

- In [BKM20], they show that a hash function $x \mapsto h_k(x) - r$ (for a random r) is CI for a (low-degree) function f by writing down an indistinguishable key distribution k_f so that $h_{k_f}(x) - f(x)$ lies in some sparse set S_f. Then, $h_{k_f}(x) - f(x) = r$ typically has no (information theoretic) solution.

- In our construction, we show that a hash function $x \mapsto h_k(x) - r$ is CI for f by writing down an indistinguishable key distribution k_f so that $h_{k_f}(x) - f(x)$ is the evaluation of a PRF $\mathsf{PRF}_s(x)$. Then, as long as it is computationally hard to find a PRF inverse $F_s^{-1}(r)$ (i.e. as long as F_s is one-way), we can conclude that the equation $h_{k_f}(x) - f(x) = r$ is computationally hard to solve.

8.2 Preliminaries

Some of the preliminaries below are adapted from [HL18, CCH19].

8.2.1 Hash Functions and Correlation Intractability

Definition 8.8. *For a pair of efficiently computable functions $(\nu(\cdot), \mu(\cdot))$, a* hash family *with input length ν and output length μ is a collection $\mathcal{H} = \{h_\lambda : \{0,1\}^{\kappa(\lambda)} \times \{0,1\}^{\nu(\lambda)} \to \{0,1\}^{\mu(\lambda)}\}_{\lambda \in \mathbb{N}}$ of keyed hash functions, along with a pair of p.p.t. algorithms:*

- $\mathcal{H}.\mathsf{Gen}(1^\lambda)$ *outputs a hash key $k \in \{0,1\}^{\kappa(\lambda)}$ describing a hash function h.*

- $\mathcal{H}.\mathsf{Hash}(k, x)$ *computes the function $h_\lambda(k, x) = h(x)$. We may use the notation $h(x)$ to denote hash evaluation when the hash family is clear from context.*

Following [HL18, CCH+19], we consider the security notion of correlation intractability [CGH98] for multi-input relations.

Definition 8.9 (Multi-Input Correlation Intractability)**.** *For a given relation ensemble* $R = \{R_\lambda \subseteq (\{0,1\}^{\nu(\lambda)})^{t(\lambda)} \times (\{0,1\}^{\mu(\lambda)})^{t(\lambda)}\}$*, a hash family* $\mathcal{H} = \{h_\lambda : \{0,1\}^{\kappa(\lambda)} \times \{0,1\}^{\nu(\lambda)} \to \{0,1\}^{\mu(\lambda)}\}$ *is said to be R-*correlation intractable with security (s, δ) *if for every s-size adversary* $\mathcal{A} = \{\mathcal{A}_\lambda\}$*,*

$$\Pr_{\substack{k \leftarrow \mathcal{H}.\mathsf{Gen}(1^\lambda) \\ \mathbf{x}=(x_1,\ldots,x_t) \leftarrow \mathcal{A}(k)}} \left[\Big(\mathbf{x}, \mathbf{y} = (h(x_1), \ldots, h(x_t))\Big) \in R\right] = O(\delta(\lambda)).$$

We say that $\mathcal{H}$ *is R-*correlation intractable with security δ *if it is* (λ^c, δ)*-correlation intractable for all* $c > 1$*. Finally, we say that* $\mathcal{H}$ *is R-*correlation intractable *if it is* $(\lambda^c, \frac{1}{\lambda^c})$*-correlation intractable for all* $c > 1$*.*

A random oracle is correlation intractable for relations that are *sparse*, defined as follows:

Definition 8.10 (Sparsity)**.** *A relation ensemble* $R = \{R_\lambda \subseteq (\{0,1\}^{\nu(\lambda)})^{t(\lambda)} \times (\{0,1\}^{\mu(\lambda)})^{t(\lambda)}\}$*, is* $\rho(\lambda)$*-sparse if for every* $\mathbf{x} \in (\{0,1\}^{\nu(\lambda)})^{t(\lambda)}$*,*

$$\Pr_{\mathbf{y} \leftarrow (\{0,1\}^{\mu(\lambda)})^{t(\lambda)}} [(\mathbf{x}, \mathbf{y}) \in R] \leq \rho(\lambda).$$

We say that R is sparse *if it is* $\mathrm{negl}(\lambda)$*-sparse.*

In this work, we focus on *distinct input relations*, i.e., relations R such that for any $(\mathbf{x}, \mathbf{y}) \in R$, we have that $x_i \neq x_j$ for any pair (i, j).

We now describe some special cases of the above definition. Two of them (CI for efficient functions and Output Intractability) have been discussed in prior works [Zha16, HL18, CCH+19, PS19], while a third – which we call "CI for shifted relations" – we introduce in this work.

Definition 8.11 (Correlation Intractability for Functions)**.** *For a given function ensemble* $\mathcal{F} = \{f_\lambda : \{0,1\}^{\nu(\lambda)} \to \{0,1\}^{\mu(\lambda)}\}$*, a hash family* $\mathcal{H} = \{h_\lambda : \{0,1\}^{\kappa(\lambda)} \times$

$\{0,1\}^{\nu(\lambda)} \to \{0,1\}^{\mu(\lambda)}\}$ *is said to be* f-correlation intractable *if it is* R*-correlation intractable for the single-input relation*

$$R = \Big\{(x, f(x)) : x \in \{0,1\}^*\Big\}.$$

Formally, the requirement is that for every poly-size $\mathcal{A} = \{\mathcal{A}_\lambda\}$*,*

$$\Pr_{\substack{k \leftarrow \mathcal{H}.\mathsf{Gen}(1^\lambda) \\ x \leftarrow \mathcal{A}(k)}} \Big[h(k,x) = f(x)\Big] = \mathrm{negl}(\lambda).$$

Definition 8.12 (Output Intractability)**.** *For a given relation ensemble* $R_{\mathrm{out}} = \{R_{\mathrm{out},\lambda} \subseteq (\{0,1\}^{\mu(\lambda)})^{t(\lambda)}\}$*, a hash family* $\mathcal{H} = \{h_\lambda : \{0,1\}^{\kappa(\lambda)} \times \{0,1\}^{\nu(\lambda)} \to \{0,1\}^{\mu(\lambda)}\}$ *is said to be* R_{out}-output intractable *if it is* R*-correlation intractable for the relation*

$$R = \Big\{(\mathbf{x}, \mathbf{y}) : \mathbf{y} \in R_{\mathrm{out}} \textit{ and } x_i \neq x_j \textit{ for all } i \neq j\Big\}.$$

Formally, the requirement is that for every poly-size $\mathcal{A} = \{\mathcal{A}_\lambda\}$*,*

$$\Pr_{\substack{k \leftarrow \mathcal{H}.\mathsf{Gen}(1^\lambda) \\ \mathbf{x} = (x_1,\ldots,x_t) \leftarrow \mathcal{A}(k)}} \Big[x_i \neq x_j \textit{ for all } i \neq j \textit{ and } \Big(\mathbf{y} = (h(x_1), \ldots, h(x_t)) \in R_{\mathrm{out}}\Big] = \mathrm{negl}(\lambda).$$

In this work, we also consider a strengthening of R_{out}-output intractability (as defined above) in which the inputs $x_1, \ldots, x_t$ are not required to be distinct; of course, this larger relation must still be sparse in order for correlation intractability to be feasible.

Definition 8.13 (Not-All-Equal (NAE) Output Intractability)**.** *For a given relation ensemble* $R_{\mathrm{out}} = \{R_{\mathrm{out},\lambda} \subseteq (\{0,1\}^{\mu(\lambda)})^{t(\lambda)}\}$*, a hash family* $\mathcal{H} = \{h_\lambda : \{0,1\}^{\kappa(\lambda)} \times \{0,1\}^{\nu(\lambda)} \to \{0,1\}^{\mu(\lambda)}\}$ *is said to be* not-all-equal R_{out}-output intractable *if it is* R*-correlation intractable for the relation*

$$R = \Big\{(\mathbf{x}, \mathbf{y}) : \mathbf{y} \in R_{\mathrm{out}} \textit{ and } x_1, \ldots, x_t \textit{ are not all equal }\Big\}.$$

When t is a constant, not-all-equal output intractability for a t-output relation

R_{out} follows from standard output intractability for $\leq t^t$ different relations defined based on R_{out} (there is one distinct-input relation for each partition of [t]). When t is superconstant it becomes better to prove the security property directly (without incurring a t^t security loss).

Definition 8.14 ((Not-All-Equal) Multi-Input CI for $\mathbb{Z}_p$-Shifted Relations)**.** *Let $p = p(\lambda)$ be an efficiently computable function of λ.*

*For a given function ensemble $\mathcal{F} = \{f_\lambda : \{0,1\}^{\nu(\lambda)} \to \mathbb{Z}_p^{\mu(\lambda)}\}$ and relation ensemble $R_{\text{out}} = \{R_{\text{out},\lambda} \subseteq (\mathbb{Z}_p^{\mu(\lambda)})^{t(\lambda)}\}$, a hash family $\mathcal{H} = \{h_\lambda : \{0,1\}^{\kappa(\lambda)} \times \{0,1\}^{\nu(\lambda)} \to \mathbb{Z}_p^{\mu(\lambda)}\}$ is said to be (R_{out}, f)-*correlation intractable *(respectively,*not-all-equal (R_{out}, f)-correlation intractable*) if it is correlation intractable for the* shifted relation

$$R = \Big\{(\mathbf{x}, \mathbf{y}) : x_i \neq x_j \textit{ for all } i \neq j \textit{ and } (y_1 - f(x_1), \ldots, y_t - f(x_t)) \in R_{\text{out}}\Big\},$$

respectively,

$$R_{\text{NAE}} = \Big\{(\mathbf{x}, \mathbf{y}) : x_1, \ldots, x_t \textit{ are not all equal } (y_1 - f(x_1), \ldots, y_t - f(x_t)) \in R_{\text{out}}.\Big\}$$

We note that Definition 8.14 generalizes both Definition 8.11 and Definition 8.12/Definition 8.13. In particular, when $p(\lambda)$ is a power-of-two, Definitions 8.12 and 8.13 can be recovered (identifying $\mathbb{Z}_p^\mu = \{0,1\}^{\mu \log p}$) by setting f to be the all-zero function, while Definition 8.11 can be recovered by setting $R_{\text{out}} = \{\mathbf{0}^\mu \in \mathbb{Z}_p^\mu = \{0,1\}^{\mu \log p}\}$.

Finally, we describe an interesting special case of Definition 8.14 that we securely instantiate under LWE.

Definition 8.15 (Weighted Sum Resistance mod p)**.** *Let $t = t(\lambda)$. A hash function family $\mathcal{H}$ with output space $\mathbb{Z}_p^\mu$ is weighted sum resistant mod p with weights $w \in \{-1, 0, 1\}^t$ if it is output intractable for the t-output relation*

$$R_{\text{out}} = \Big\{\mathbf{y} : \sum_{i=1}^{t} w_i y_i = 0^\mu \ \textit{(mod p)}\Big\}.$$

Similarly, it is not-all-equal weighted sum resistant mod p with weights w if it is

NAE output intractable for R_{out}.

We say that $\mathcal{H}$ is $\mathsf{weighted\ sum\ resistant}$ if it is sum resistant for all nonzero weight vectors w, and $\mathsf{NAE\text{-}weighted\ sum\ resistant}$ if it is NAE-sum resistant for all weight vectors w such that $\sum_i w_i \neq 0$. As shown in Section 8.4, our LWE-based hash family satisfies (NAE) multi-input CI for (both variants of) *shifted* weighted sum resistance mod p with $p \approx 2^\lambda$.

8.2.2 Shift-Hiding Shiftable Functions

We consider a weakening of the original definition of Peikert and Shiehian [PS18] that does not give the adversary oracle access to the SHSF. We also consider a modified definition with exact correctness rather than approximate correctness (this corresponds to the "rounded version" of the [PS18] construction).

Definition 8.16 (Shift-Hiding Shiftable Functions [PS18])**.** *Let $p = p(\lambda)$ be an efficiently computable function of λ. We define a family of shift-hiding shiftable functions with input space $\{0,1\}^{\nu(\lambda)}$ and output space $\mathbb{Z}_p^{\mu(\lambda)} = \{0,1\}^{\mu(\lambda)\log p(\lambda)}$ for arbitrary polynomial functions $(\nu(\lambda), \mu(\lambda))$.*

For a given class $\mathcal{C}$ of function ensembles $\mathcal{F} = \{f_\lambda : \{0,1\}^{\nu(\lambda)} \to \mathbb{Z}_p^{\mu(\lambda)}\}$, a $\mathsf{shift\text{-}hiding\ shiftable\ function\ family}$ $\mathsf{SHSF} = (\mathsf{Gen}, \mathsf{Shift}, \mathsf{Eval}, \mathsf{SEval})$ *consists of four PPT algorithms:*

- $\mathsf{Gen}(1^\lambda)$ *outputs a master secret key* msk *and public parameters* pp.
- $\mathsf{Shift}(\mathsf{msk}, f)$ *takes as input a secret key* msk *and a function $f \in \mathcal{F}$. It outputs a shifted key* sk_f.
- $\mathsf{Eval}(\mathsf{pp}, \mathsf{msk}, x)$*, given a secret key* msk *and input $x \in \{0,1\}^{\nu(\lambda)}$, outputs an evaluation $y \in \mathbb{Z}_p^{\mu(\lambda)}$.*
- $\mathsf{SEval}(\mathsf{pp}, \mathsf{sk}_f, x)$*, given a shifted key* sk_f *and input $x \in \{0,1\}^{\nu(\lambda)}$, outputs an evaluation $y \in \mathbb{Z}_p^{\mu(\lambda)}$.*

We will sometimes use the notation $F_{\mathsf{sk}}(x)$ to mean either $\mathsf{Eval}(\mathsf{sk}, x)$ or $\mathsf{SEval}(\mathsf{sk}, x)$ when the context is clear.

We require that SHSF *satisfies the following two properties:*

- ***Computational Correctness**: for any function $f \in \mathcal{C}$, given public parameters* pp *and a shifted key $\mathsf{sk}_f \leftarrow \mathsf{Shift}(\mathsf{msk}, f)$ (for $(\mathsf{pp}, \mathsf{msk}) \leftarrow \mathsf{Gen}(1^\lambda)$), it is computationally hard to find an input $x \in \{0,1\}^{\nu(\lambda)}$ such that $\mathsf{Eval}(\mathsf{sk}_f, x) \neq \mathsf{Eval}(\mathsf{msk}, x) + f(x)$ (mod p). In other words, the equation*

$$F_{\mathsf{sk}_f}(x) = F_{\mathsf{msk}}(x) + f(x)$$

  holds computationally (mod p).

- ***Shift Hiding**: for any pair of functions $f, g \in \mathcal{C}$,*

$$\mathsf{sk}_f \approx_c \mathsf{sk}_g,$$

  where $\mathsf{sk}_f \leftarrow \mathsf{Shift}(\mathsf{msk}, f)$, $\mathsf{sk}_g \leftarrow \mathsf{Shift}(\mathsf{msk}, g)$, and $\mathsf{msk} \leftarrow \mathsf{Gen}(1^\lambda)$.

8.2.3 Learning with Errors and (One-Dimensional) Short Integer Solution

We begin with definitions of the learning with errors (LWE) and sort integer solution (SIS) problems, following Peikert's survey [Pei16]. We refer the reader to [Pei16] for definitions of worst-case lattice problems such as SIVP and GapSVP.

Definition 8.17 (Learning with Errors)**.** *For integers $n, m, q \in \mathbb{N}$ and error distribution χ, the learning with errors problem $\mathsf{LWE}_{n,m,q,\chi}$ is defined to be the following average-case decision problem: distinguish between a uniformly random matrix-vector pair*

$$(\mathbf{A} \leftarrow \mathbb{Z}_q^{n\times m}, \mathbf{u} \leftarrow \mathbb{Z}_q^m)$$

and an approximate linear equation

$$(\mathbf{A} \leftarrow \mathbb{Z}_q^{n\times m}, \mathbf{sA} + \mathbf{e})$$

with $\mathbf{s} \leftarrow \mathbb{Z}_q^n$ *and* $\mathbf{e} \leftarrow \chi^m$*.*

Definition 8.18 (Short Integer Solution)**.** *For integers* $n, m, q, B \in \mathbb{N}$*, the short integer solution problem* $\mathsf{SIS}_{n,m,q,B}$ *is defined to be the following search problem: given a uniformly random matrix*

$$\mathbf{A} \leftarrow \mathbb{Z}_q^{n\times m}$$

find a vector $\mathbf{v} \in \mathbb{Z}_q^m$ *such that* $||\mathbf{v}||_\infty \leq B$ *and* $\mathbf{Av} = 0^n$*.*

One-Dimensional SIS Variants

We also explicitly consider two different "one-dimensional" variants of SIS that come up in our security proofs. One variant is the "1D-R-SIS problem" as defined by [BV15]; the other is a variant implicitly considered by [Ajt96] and explicitly considered by [Reg04, BV15] that we will call "(approximate) $\mathbb{Z}_q$-SIS."[9]

These problems are no easier to solve than LWE, but for clarity, as was done in [BV15, PS18], it is convenient to define them separately.

Definition 8.19 (Approximate $\mathbb{Z}_q$-SIS)**.** *For positive integers* $q, m, B, E \in \mathbb{N}$*, the approximate* $\mathbb{Z}_q$*-SIS problem is defined as follows: given a uniformly random vector* $\mathbf{v} \in \mathbb{Z}_q^m$*, find a nonzero vector* $\mathbf{z} \in \mathbb{Z}^m$ *such that:*

- $||\mathbf{z}||_\infty \leq B$*; and*
- $\langle \mathbf{v}, \mathbf{z}\rangle \pmod{q} \in [-E, E]$.

In [Reg04, BV15], it is shown that $\mathbb{Z}_q$-SIS is as hard as worst-case lattice problems in the following parameter regime (among others):

[9] The problem called "1D-SIS" in [BV15] is a special case of approximate $\mathbb{Z}_q$-SIS; the two error parameters (B, E) in Definition 8.19 below are set to be equal to each other in [BV15].

Fact 8.20. *If $q = \prod_{i=1}^n p_i$ and each $p_i \geq B \cdot \omega(\sqrt{mn\log(n)})$, then $\mathbb{Z}_q$-$\mathsf{SIS}_{m,q,B,E=B}$ is as hard as $\mathsf{SIVP}_{B\cdot\tilde{O}(\sqrt{mn})}$ and $\mathsf{GapSVP}_{B\cdot\tilde{O}(\sqrt{mn})}$ for n-dimensional lattices.*

However, we will be interested in a variant of approximate $\mathbb{Z}_q$-SIS where E is very large compared to B; therefore, we appeal to a simple modulus switching [BV11] reduction.

Claim 8.20.1. *The approximate $\mathbb{Z}_q$-SIS problem with parameters (q, m, β, η) reduces to the approximate $\mathbb{Z}_Q$-SIS problem with parameters (Q, m, B, E) if $\beta \geq B$ and*

$$\frac{\eta}{q} \geq \frac{E}{Q} + \frac{mB}{Q} + \frac{mB}{q}$$

We will invoke this claim (see Section 8.4.5) in a setting where $Q \gg q$ (in fact, we will set $q \ll \frac{Q}{E}$ so that the first term in this sum is insignificant).

Proof. Given $\mathbf{v} \in \mathbb{Z}_q^m$ (interpreted as an integer vector), define $\mathbf{V} \in \mathbb{Z}_Q^m$ so that each coordinate satisfies $V_i = \left\lceil \frac{Q}{q} v_i + r_i \right\rfloor$, where r_i is a uniformly random real number in the range $[0, \frac{Q}{q}]$. We then have that

$$\mathbf{V} = \frac{Q}{q}\mathbf{v} + \epsilon$$

for a vector $\epsilon \in \mathbb{R}^m$ such that $||\epsilon||_\infty \leq 1 + \frac{Q}{q}$. Note that $\mathbf{V}$ is a uniformly random element of $\mathbb{Z}_Q^m$, so the reduction is valid. Now, assuming that the $\mathbb{Z}_Q$-SIS problem is solved correctly, we are given a vector $\mathbf{z}$ such that

$$\langle \mathbf{V}, \mathbf{z} \rangle = Q \cdot \ell + e$$

and $|e| \leq E$. Then,

$$\langle \mathbf{v}, \mathbf{z} \rangle = q\ell + \frac{q}{Q}e - \frac{q}{Q}\langle \epsilon, \mathbf{z} \rangle,$$

which breaks approximate $\mathbb{Z}_q$-SIS with parameters (q, m, B, η) as long as

$$\eta \geq \frac{q}{Q}E + m\frac{q}{Q}(1 + \frac{Q}{q})B$$

$$= \frac{q}{Q}E + \frac{q}{Q} \cdot mB + mB \qquad \square$$

In addition to approximate $\mathbb{Z}_q$-SIS, we consider a slight variant of 1D-R-SIS [BV15] due to [BKM17].

Definition 8.21 (1D-R-SIS [BV15, BKM17])**.** *Let $p \in \mathbb{N}$ and $p_1 < p_2 < \ldots < p_n$ be pairwise coprime and relatively prime to p. Let $q = p \cdot \prod_{i=1}^{n} p_i$. Then, for positive integers $m \in \mathbb{N}$ and B, the $\mathsf{1D\text{-}R\text{-}SIS}_{m,p,q,B}$ problem is as follows: given a uniformly random vector $\mathbf{v} \in \mathbb{Z}_q^m$, find a nonzero vector $\mathbf{z} \in \mathbb{Z}^m$ such that*

- *$||\mathbf{z}||_\infty \leq B$; and*
- *$\langle \mathbf{v}, \mathbf{z} \rangle \pmod{q} \in \frac{q}{p} \cdot (\mathbb{Z} + \frac{1}{2}) + [-B, B]$.*

Fact 8.22. *([Ajt96, BV15, BKM17]) For sufficiently large $p_i \geq B \cdot \mathsf{poly}(n, \log q)$, solving 1D-R-SIS is at least as hard as approximating SIVP and SVP on arbitrary n-dimensional lattices to within $B \cdot \mathsf{poly}(n)$ factors.*

8.3 Correlation Intractability from Shift-Hiding Shiftable Functions

In this section, we show that shift-hiding shiftable functions (Definition 8.16) that are *output intractable* (Definitions 8.12 and 8.13) can be used to construct correlation-intractable hash functions for shifted relations (Definition 8.14). As a special case, this shows that SHSFs that are *hard to invert* yield correlation-intractable hash functions for all circuits (Definition 8.11) supported by the SHSF function class $\mathcal{C}$. In other words, SHSFs allow us to *lift* a form of output intractability to a more general form of correlation intractability.

Formally, let $\mathsf{SHSF} = (\mathsf{Gen}, \mathsf{Shift}, \mathsf{Eval})$ be a SHSF family that represents functions of the form $F_{\mathsf{sk}} : \{0,1\}^{\nu(\lambda)} \to \mathbb{Z}_p^{\mu(\lambda)}$ and supports shifts for functions $f \in \mathcal{C}$, where $\mathcal{C}$ is some class that contains the all zero function ensemble. We then consider two hash functions $\mathcal{H}_{\text{plain}}, \mathcal{H}_{\text{shift}}$:

- $\mathcal{H}_{\text{plain}}$ uses msk as a hash key, and computes the function $h(\mathsf{msk}, x) = F_{\mathsf{msk}}(x)$.

- $\mathcal{H}_{\text{shift}}$ uses sk_Z as a hash key, where $Z : \{0,1\}^\nu \to \mathbb{Z}_p^\mu$ is an identically zero function. It computes the function $h(\mathsf{sk}_Z, x) = F_{\mathsf{sk}_Z}(x)$.

Theorem 8.23. *Let R_{out} be an efficiently decidable output relation. If* SHSF *is a shift-hiding shiftable function family for $\mathcal{C}$ and $\mathcal{H}_{\text{plain}}$ is R_{out}-output intractable, then $\mathcal{H}_{\text{shift}}$ is (R, f)-correlation intractable for any $f \in \mathcal{C}$.*

Moreover, if $\mathcal{H}_{\text{plain}}$ is NAE-R_{out}-output intractable, then $\mathcal{H}_{\text{shift}}$ is NAE-(R, f)-CI for any $f \in \mathcal{C}$.

Proof. Suppose that a PPT adversary $\mathcal{A}$ breaks the (R, f)-correlation intractability of $\mathcal{H}_{\text{shift}}$, which means that $\mathcal{A}$ wins the following challenger-based security game with non-negligible probability:

1. The challenger samples $\mathsf{msk} \leftarrow \mathsf{Gen}(1^\lambda)$.

2. The challenger samples $\mathsf{sk} = \mathsf{sk}_Z \leftarrow \mathsf{Shift}(\mathsf{msk}, Z)$ and sends sk to $\mathcal{A}$.

3. $\mathcal{A}(\mathsf{sk})$ outputs $\mathbf{x} = (x_1, \ldots, x_t)$.

4. $\mathcal{A}$ wins if (i) the inputs x_i are distinct, and (ii) for $y_i = F_{\mathsf{sk}}(x_i) - f(x_i)$, the relation $R_{\text{out}}(\mathbf{y})$ holds.

Then, $\mathcal{A}$ also wins each of the following modified security games with non-negligible probability.

- Hybrid Hyb_1: same as the honest security game, except that in step (2), we sample

$$\mathsf{sk}_f \leftarrow \mathsf{Shift}(\mathsf{msk}, f)$$

  This is indistinguishable from the original security game by the shift-hiding of SHSF.

- Hybrid Hyb_2: same as Hyb_1, except that in step (4), we change the win condition (ii) so that $\mathcal{A}$ wins if for $y_i = F_{\mathsf{msk}}(x_i)$, the relation $R_{\text{out}}(\mathbf{y})$ holds.

  This is indistinguishable from Hyb_1 by the computational correctness of SHSF.

Finally, we show that $\mathcal{A}$'s success in Hyb_2 leads to an attack $\mathcal{A}'$ on the R_{out}-output intractability of $\mathcal{H}_{\text{plain}}$. The attack works as follows:

1. The challenger samples $\mathsf{msk} \leftarrow \mathsf{Gen}(1^\lambda)$ and sends msk to $\mathcal{A}'$.
2. $\mathcal{A}'(\mathsf{msk})$ samples $\mathsf{sk} = \mathsf{sk}_f \leftarrow \mathsf{Shift}(\mathsf{msk}, f)$.
3. $\mathcal{A}'$ then calls $\mathcal{A}(\mathsf{sk}_f)$ and outputs $\mathbf{x} = (x_1, \dots, x_\ell)$.
4. By definition, $\mathcal{A}'$ wins if (i) the x_i are distinct, and (ii) for $y_i = F_{\mathsf{msk}}(x_i)$, the relation $R_{\text{out}}(\mathbf{y})$ holds.

By construction, $\mathcal{A}'$ above wins with the same probability that $\mathcal{A}$ wins in Hyb_2, contradicting the R_{out}-output intractability of $\mathcal{H}_{\text{plain}}$.

The same argument as above applies to NAE-CI, with the condition (i) replaced by "the inputs x_i are not all equal." This completes the proof of Theorem 8.23. □

8.4 Construction of (Weighted) Sum-Resistant SHSF

We show the (weighted) sum-resistance of a variant of the Peikert-Shiehian construction of shift-hiding shiftable functions [PS18]. We start by describing the ingredients that we use in the construction; the construction itself is described in Section 8.4.2. We include proof sketches of computational correctness in Section 8.4.3 and shift-hiding in Section 8.4.4 for completeness, although these follow the original [PS18] result. Finally, the proof of sum-resistance (which is new to this work) is in Section 8.4.5. Appropriate parameter balancing must be done to ensure that the three security reductions are simultaneously valid for a single set of parameters.

8.4.1 The Ingredients

The Gadget Matrix. An important ingredient in many lattice-based constructions is the gadget matrix $\mathbf{G}$ and the operator $\mathbf{G}^{-1}$ associated to it. Let

$$\mathbf{g} = [1,\ 2,\ 4,\ \dots, 2^{\lceil \log q \rceil - 1}] \in \mathbb{Z}_q^{1 \times \lceil \log q \rceil}$$

The gadget matrix $\mathbf{G} = \mathbf{I}_n \otimes \mathbf{g}$ is a block diagonal matrix with copies of $\mathbf{g}$ on the diagonal. In fact, we will extend $\mathbf{G}$ to m columns for any $m \geq n\lceil \log q \rceil$ by appending zero columns.

An important property of $\mathbf{G} \in \mathbb{Z}_q^{n \times m}$ is that for every vector $\mathbf{v} \in \mathbb{Z}_q^n$, there is a 0-1 vector $\mathbf{v}' \in \{0,1\}^m$ such that $\mathbf{G}\mathbf{v}' = \mathbf{v} \pmod{q}$. This leads us to define the operator $\mathbf{G}^{-1} : \mathbb{Z}_q^n \rightarrow \{0,1\}^m$ which has the property that

1. $\mathbf{G}^{-1}(\mathbf{v}) \in \{0,1\}^m$ for every vector $\mathbf{v} \in \mathbb{Z}_q^n$; and

2. $\mathbf{G} \cdot \mathbf{G}^{-1}(\mathbf{v}) = \mathbf{v} \pmod{q}$.

We will extend $\mathbf{G}^{-1}$ to matrices $\mathbf{V}$ by acting on each column of the matrix separately. We caution the reader that $\mathbf{G}^{-1}$ refers to a (non-linear) operator, and has little to do with matrix inverses.

Gadget Homomorphisms. The key idea in the SHSF construction is the notion of gadget homomorphisms originating from [BGG+14]. For LWE matrices $\mathbf{A}_1, \mathbf{A}_2 \in \mathbb{Z}_q^{n \times m}$, define the sum and product matrices

$$\mathbf{A}_+ = \mathbf{A}_1 + \mathbf{A}_2 \;\text{ and }\; \mathbf{A}_\times = -\mathbf{A}_1 \mathbf{G}^{-1}(\mathbf{A}_2) \tag{8.1}$$

where $\mathbf{G}$ is the gadget matrix and $\mathbf{G}^{-1}$ is the bit decomposition operator defined above. The gadget homomorphisms allow us to start from LWE encodings $\mathbf{c}_1 \approx \mathbf{s}(\mathbf{A}_1 + x_1\mathbf{G})$ and $\mathbf{c}_2 \approx \mathbf{s}(\mathbf{A}_2 + x_2\mathbf{G})$ w.r.t. an LWE secret $\mathbf{s} \in \mathbb{Z}_q^n$ (where we suppress the LWE errors for clarity) and compute

$$\mathbf{c}_+ \approx \mathbf{s}(\mathbf{A}_+ + (x_1 + x_2)\mathbf{G}) \;\text{ and }\; \mathbf{c}_\times \approx \mathbf{s}(\mathbf{A}_\times + x_1 x_2 \mathbf{G}) \tag{8.2}$$

In particular, this is accomplished by setting

$$\mathbf{c}_+ = \mathbf{c}_1 + \mathbf{c}_2 \approx \mathbf{s}(\mathbf{A}_1 + \mathbf{A}_2 + (x_1 + x_2)\mathbf{G}) = \mathbf{s}(\mathbf{A}_+ + (x_1 + x_2)\mathbf{G})$$

and

$$\begin{aligned}\mathbf{c}_\times &= -\mathbf{c}_1\mathbf{G}^{-1}(\mathbf{A}_2) + x_1\mathbf{c}_2\\ &\approx -\mathbf{s}(\mathbf{A}_1 + x_1\mathbf{G}) \cdot \mathbf{G}^{-1}(\mathbf{A}_2) + \mathbf{s}(\mathbf{A}_2 + x_2\mathbf{G}) \cdot x_1\\ &= \mathbf{s}(-\mathbf{A}_1\mathbf{G}^{-1}(\mathbf{A}_2) + x_1x_2\mathbf{G})\\ &= \mathbf{s}(\mathbf{A}_\times + x_1x_2\mathbf{G})\end{aligned}$$

Crucially, this computation does not require the knowledge of either x_1 or x_2 to compute the sum. It does require the knowledge of x_1 (but not x_2) to compute the product. This *asymmetry* will prove invaluable to us down the line. We will ensure that the inputs x_i as well as the intermediate values in the computation are bits, in order to control the error growth.

More generally, we define the following two algorithms.

- $\mathsf{Gadget.MEval}(g, \mathbf{A}_1, \ldots, \mathbf{A}_\ell)$, the matrix homomorphism, takes as input a function $g : \{0,1\}^\ell \to \{0,1\}$ and ℓ matrices $\mathbf{A}_1, \ldots, \mathbf{A}_\ell$ and outputs the matrix $\mathbf{A}_g$ obtained by composing together the addition and multiplication operations in Equation 8.1.

- $\mathsf{Gadget.VEval}(g, x, \mathbf{c}_1, \ldots, \mathbf{c}_\ell)$, the vector homomorphism, takes as input a function $g : \{0,1\}^\ell \to \{0,1\}$, an input $x = x_1x_2 \ldots x_\ell$ and LWE encodings

$$\mathbf{c}_1 = \mathbf{s}(\mathbf{A}_1 + x_1\mathbf{G}) + \mathbf{e}_1, \ldots, \mathbf{c}_\ell = \mathbf{s}(\mathbf{A}_\ell + x_\ell\mathbf{G}) + \mathbf{e}_\ell$$

  of x w.r.t. $\mathbf{A}_1, \ldots, \mathbf{A}_\ell$, and outputs a vector $\mathbf{c}_g$ obtained by composing together the addition and multiplication operations in Equation 8.2.

Correctness tells us that if $\mathbf{c}_1, \ldots, \mathbf{c}_\ell$ have $\mathsf{poly}(n)$-bounded error, then

$$\mathbf{c}_g \approx \mathbf{s}(\mathbf{A}_g + g(x)\mathbf{G}) \tag{8.3}$$

where the difference is an LWE error whose magnitude is $O((n \log q)^{O(d_g)})$ where λ is a security parameter and d_g is the depth of the circuit g. Looking ahead, we make two important observations on these algorithms:

1. First, if the function g is of a special form, namely $g(x_1, x_2) = \langle x_1, f(x_2)\rangle$ for some $x = x_1||x_2$, then $\mathsf{Gadget.VEval}$ does not require the knowledge of x_1, rather only x_2. Furthermore, while we required all the numbers in a computation to be bits so far, a terminal inner product (i.e. an inner product at the end of a computation) can support x_1 being a vector consisting of large numbers. These observations are due to [AFV11, GVW15] where they were used to construct a predicate encryption scheme.

2. Secondly, if the first coordinate of $\mathbf{s}$ is 1 (which we can set without loss of security) then we have

$$c_g \approx \mathsf{s}\mathsf{a}_g + g(x) \tag{8.4}$$

   where c_g is the first coordinate of $\mathbf{c}_g$ and a_g is the first column of $\mathbf{A}_g$. This is because the first column of $\mathbf{G}$ is the unit vector with 1 in the first coordinate and 0 everywhere else.

FHE with Almost Linear Decryption. We require the existence of a (secret-key) FHE scheme where the secret key fsk is a vector $\mathbf{s} \in \mathbb{Z}_q^{\widehat{n}}$, ciphertexts fct of messages $m \in \mathbb{Z}_p$ are vectors $\mathbf{c} \in \mathbb{Z}_q^{\widehat{n}}$ and decryption proceeds by first doing a linear operation which gives

$$\langle \mathsf{fsk}, \mathsf{fct}\rangle = m \cdot \left\lfloor \frac{q}{p} \right\rceil + e \pmod q \tag{8.5}$$

where e is a small error. In particular, we will ask that if initial ciphertexts have polynomially bounded error, then $||e||$ should be bounded by $(\hat{n}\log q)^{O(d)}$, where d is the depth of the homomorphic computation. Prior LWE-based FHE schemes, as constructed in [BV11,BGV12,GSW13,BV14,AP14], have this form (based on different variants of LWE). We will let $\mathsf{FHE.Enc}$ denote the encryption algorithm and $\mathsf{FHE.Eval}$ denote the evaluation algorithm.

8.4.2 The Shift-Hiding Shiftable Function

Let the class of functions $\mathcal{C}$ consist of functions $f : \{0,1\}^\nu \to \mathbb{Z}_p^\mu$ computable by circuits of size at most $s = s(\lambda)$. We require that $p = p(\lambda)$ is a sufficiently large function of λ; for simplicity, we will choose p so that $p = 2^{\Theta(\lambda)}$ (further specified later). Since we allow $\mu(\lambda), \nu(\lambda)$ to be arbitrary polynomial functions, every function family with polynomially related input and output length can be expressed in such a way.

- $\mathsf{Gen}(1^\lambda)$: picks LWE parameters $n = n(\lambda)$, $m = m(\lambda)$ and $q = q(\lambda)$, where $q = 2^{(d\lambda)^{O(1/\epsilon)}}$ is a sufficiently large product of primes to be specified later. We pick the LWE error distribution to be polynomially bounded, and set $n \geq (d\lambda)^{O(1/\epsilon^2)}$ so that the LWE assumption follows from worst-case hardness of GapSVP with subexponential approximation factors. Generate the public parameters

  $$\mathsf{pp} = (\mathbf{A}_1, \ldots, \mathbf{A}_\ell, \mathbf{u}) \leftarrow (\mathbb{Z}_q^{n\times m})^\ell \times \mathbb{Z}_q^{1\times m}$$

  for a certain $\ell = \ell(s, \lambda)$ (also specified later).

  Choose a uniformly random vector $\mathbf{s} \leftarrow \mathbb{Z}_q^n$ whose first coordinate $\mathbf{s}[1] = 1$. Let $\mathsf{msk} = \mathbf{s}$.

- $\mathsf{Eval}(\mathsf{msk}, x)$: Let FHE be a (leveled) fully homomorphic encryption scheme with almost linear decryption (as defined above in Equation 8.5) with plaintext space $\mathbb{Z}_p$. Construct the functions $g_x^{(i)}$ that, on input a pair $(\mathsf{fsk}, \mathsf{fct})$, output[10]

  $$g_x^{(i)}(\mathsf{fsk}, \mathsf{fct}) = \left\langle \mathsf{fsk}, \mathsf{FHE.Eval}(\mathsf{fct}, \mathcal{U}_x^{(i)}) \right\rangle \pmod q$$

  where $\mathcal{U}_x^{(i)}$ is a universal circuit that takes as input the description of a circuit f and outputs the i^{th} $\mathbb{Z}_p$-block of $f(x)$. The parameter $\ell = \mathsf{poly}(\nu, \mu, \lambda)$ is set to be large enough so that the functions $g_x^{(i)}$ have description length at most ℓ.

[10] The function $g_x^{(i)}$ does not actually have a binary output, but as was done in [BV15, GVW15], the [BGG+14] homomorphism can be extended to this function.

Define

$$\mathbf{A}_x^{(i)} = \mathsf{Gadget.MEval}(g_x^{(i)}, \mathbf{A}_1, \ldots, \mathbf{A}_\ell) \in \mathbb{Z}_q^{n \times m} ,$$

let $\mathsf{a}_x^{(i)}$ denote the first column of $\mathbf{A}_x^{(i)}$ and let

$$\mathbf{A}_x := [\mathsf{a}_x^{(1)} || \mathsf{a}_x^{(2)} || \ldots || \mathsf{a}_x^{(\mu)}] \in \mathbb{Z}_q^{n \times \mu}$$

denote the concatenation of $\mathsf{a}_x^{(i)}$. The output is

$$\lfloor \mathbf{s}\mathbf{A}_x + \mathbf{u}\mathbf{G}^{-1}(\mathbf{A}_x) \rceil_p := \left\lfloor \frac{p}{q} \cdot (\mathbf{s}\mathbf{A}_x + \mathbf{u}\mathbf{G}^{-1}(\mathbf{A}_x)) \right\rceil \in \mathbb{Z}_p^{1 \times \mu}$$

- $\mathsf{Shift}(\mathsf{msk}, f)$: Choose an FHE secret key $\mathsf{fsk} \in \mathbb{Z}_q^{\widehat{n}}$, encrypt the description of f into an FHE ciphertext fct, let $\phi := \mathsf{fct} || \mathsf{fsk}$, and let

  $$\mathbf{A}_f := [\mathbf{A}_1 + \phi_1 \mathbf{G} || \ldots || \mathbf{A}_\ell + \phi_\ell \mathbf{G}]$$

  Output as the shift key

  $$\mathsf{sk}_f := (\mathsf{fct}, \mathbf{s}\mathbf{A}_f + \mathbf{e})$$

  where $\mathbf{e}$ is drawn from the LWE noise distribution.

  Note that ℓ is the bit-length of $\mathsf{fsk} || \mathsf{fct}$ and is $\mathsf{poly}(s, \lambda)$.

- $\mathsf{SEval}(\mathsf{sk}_f, x)$: Let the circuits $g_x^{(i)}$ be as in the definition of Eval.

  $$\mathbf{c}_x^{(i)} = \mathsf{Gadget.VEval}(g_x^{(i)}, \mathsf{fct}, \mathbf{c}_1, \ldots, \mathbf{c}_\ell) \in \mathbb{Z}_q^n$$

  where $\mathbf{c}_i = \mathbf{s}[\mathbf{A}_i + \phi_i \mathbf{G}]$. Note that crucially, $\mathsf{Gadget.VEval}$ does not require fsk as input because, by observation (1) above, $g_x^{(i)}$ only linearly depends on it. Let $c_x^{(i)}$ denote the first element of $\mathbf{c}_x^{(i)}$ and let $\mathbf{c}_x$ be the concatenation of all $c_x^{(i)}$.

  Output

  $$\lfloor \mathbf{c}_x + \mathbf{u}\mathbf{G}^{-1}(\mathbf{A}_x) \rceil_p$$

  as the shifted evaluation.

8.4.3 Proof of Computational Correctness

Computational correctness follows from a similar argument in [PS18], although with slightly different parameter choices. We sketch it here for completeness.

Basic Correctness. We first sketch correctness of SEval for any fixed x. By the correctness of the gadget homomorphisms (equation 8.4), we know that

$$\begin{aligned} c_x^{(i)} &\approx \mathsf{sa}_x^{(i)} + g_x^{(i)}(\mathsf{fsk}, \mathsf{fct}) \\ &= \mathsf{sa}_x^{(i)} + \langle \mathsf{fsk}, \mathsf{FHE.Eval}(\mathsf{fct}, \mathcal{U}_x^{(i)})\rangle \\ &\approx \mathsf{sa}_x^{(i)} + \mathcal{U}_x^{(i)}(f) \cdot \left\lfloor \frac{q}{p} \right\rceil \\ &= \mathsf{sa}_x^{(i)} + f^{(i)}(x) \cdot \left\lfloor \frac{q}{p} \right\rceil \end{aligned} \tag{8.6}$$

where the second equation is by the definition of $g_x^{(i)}$, the third (approximate) equation is by the correctness of FHE decryption (equation 8.5), and the fourth equation is by the definition of the universal circuit $\mathcal{U}$. The approximation error is equal to the gadget homomorphic evaluation error plus the FHE decryption error, which is

$$O((n \log q)^{O(d')} + (\hat{n} \log q)^{O(d)}) = \lambda^{O(\frac{1}{\epsilon^4} \cdot d \log(d\lambda))}$$

where d is the depth of the circuit $\mathcal{U}_x^{(i)}$ and $d' = O(d \cdot \log(n \log q))$ is the depth of the circuit $g_x^{(i)}$ that homomorphically evaluates $\mathcal{U}_x^{(i)}$ and decrypts. Since we chose $q = 2^{\lambda^{\Theta(1/\epsilon)}}$, this error is very small relative to q.

Now, as long as $c_x^{(i)}$ does not fall too close to the boundaries of multiples of q/p, we have

$$\begin{aligned} \mathsf{SEval}(\mathsf{sk}_f, x) &= \lfloor \mathbf{c}_x + \mathbf{u}\mathbf{G}^{-1}(\mathbf{A}_x) \rceil_p \\ &= \lfloor \mathbf{s}\mathbf{A}_x + \mathbf{u}\mathbf{G}^{-1}(\mathbf{A}_x) \rceil_p + f(x) = \mathsf{Eval}(\mathsf{msk}, x) + f(x) \pmod p \end{aligned} \tag{8.7}$$

It turns out that for any fixed x, the boundary event happens with a negligible probability. Moreover, adapting arguments from [BV15,PS18], we will now show that it is computationally hard to find an x for which correctness (that is, equation 8.7) fails. (This is stronger than basic correctness in that it holds for any adaptively chosen x, and weaker because the guarantee is computational; adaptive statistical correctness does not hold for this construction.)

Computational Correctness. By the calculation in equation 8.6, we know that for each $i \in [\mu]$,

$$c_x^{(i)} = \mathsf{sa}_x^{(i)} + f^{(i)}(x) \cdot \left\lfloor \frac{q}{p} \right\rceil + e_i$$

where $|e_i| \le B := \lambda^{O(\frac{1}{\epsilon^4} d \log(d\lambda))}$.

Assume that there is an adversary that, given the shift key $\mathsf{sk}_f \leftarrow \mathsf{Shift}(\mathsf{msk}, f)$ for some f of his choice, produces an x such that

$$\mathsf{SEval}(\mathsf{sk}_f, x) \neq \mathsf{Eval}(\mathsf{msk}, x)$$

meaning that they differ in some coordinate, say i.

Then, by the expressions for SEval and Eval, we have

$$\begin{aligned}
\mathsf{SEval}(\mathsf{sk}_f, x)|_i = \left\lfloor \frac{p}{q} c_x^{(i)} \right\rceil &= \left\lfloor \frac{p}{q} \cdot (\mathsf{sa}_x^{(i)} + f^{(i)}(x) \cdot \left\lfloor \frac{q}{p} \right\rceil + e_i) \right\rceil \\
&= \left\lfloor \frac{p}{q} \cdot (\mathsf{sa}_x^{(i)} + f^{(i)}(x) \cdot \frac{q}{p} + e'_i) \right\rceil \\
&\neq \left\lfloor \frac{p}{q} \cdot (\mathsf{sa}_x^{(i)} + f^{(i)}(x) \cdot \frac{q}{p}) \right\rceil \\
&= \left\lfloor \frac{p}{q} \cdot \mathsf{sa}_x^{(i)} \right\rceil + f^{(i)}(x) = \mathsf{Eval}(\mathsf{msk}, x)|_i
\end{aligned}$$

where $\epsilon'_i = \epsilon_i + f^{(i)}(x) \left(\left\lfloor \frac{q}{p} \right\rceil - \frac{q}{p} \right) \in [-(B+p), (B+p)]$. This can only happen when

$$\frac{p}{q} c_x^{(i)} \in \mathbb{Z} + \frac{1}{2} + \frac{p}{q} \cdot [-(B+p), B+p],$$

or, equivalently,

$$c_x^{(i)} \in \frac{q}{p}(\mathbb{Z} + \frac{1}{2}) + [-(B+1/2), B+1/2].$$

Now, observe that

$$c_x^{(i)} = [\mathbf{c}_1|| \dots ||\mathbf{c}_\ell] \cdot \mathbf{h}^{(i)}$$

for some vector $\mathbf{h}^{(i)}$ of low norm $B = \lambda^{O(\frac{1}{\epsilon^4} d \log(d\lambda))}$. Since $\mathbf{c}_i$ are pseudorandom, this gives us a solution to the $\mathsf{1D\text{-}SIS}_{\ell,p,q,B}$ problem. For this choice of B, Fact 8.22 tells us that provided $q = p \prod_{i=1}^{n'} p_i$ such that each $p_i \geq \mathsf{poly}(B)$, this 1D-SIS variant is as hard as $\mathsf{SIVP}/\mathsf{GapSVP}$ on n'-dimensional lattices with an approximation factor of $\mathsf{poly}(B)$. Given all of the parameter constraints, we can set $n' \geq (d\lambda)^{O(1/\epsilon)}$ so that $2^{(n')^\epsilon} \geq \mathsf{poly}(B)$, allowing us to rely on the claimed hardness assumption.

8.4.4 Proof of Shift-Hiding

We wish to show that for any two functions $f_0, f_1 \in \mathcal{C}$,

$$(\mathsf{Shift}(\mathsf{msk}, f_0), \mathsf{pp}) \approx_c (\mathsf{Shift}(\mathsf{msk}, f_1), \mathsf{pp})$$

where $(\mathsf{pp}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda)$. This also follows from [PS18] (up to minor definition/notation changes), but we sketch a proof for completeness. Shift-hiding follows by the following sequence of hybrids.

Hybrid 0. This is the distribution generated by picking $(\mathsf{pp}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda)$ and outputting pp together with

$$\mathsf{sk}_{f_0} \leftarrow \mathsf{Shift}(\mathsf{msk}, f_0)$$

That is,

$$\mathsf{sk}_{f_0} := (\mathsf{fct}, \mathbf{s}\mathbf{A}_{f_0} + \mathbf{e})$$

where fct is an FHE encryption of f_0 under an FHE secret key fsk, and

$$\mathbf{A}_{f_0} = [\mathbf{A}_1 + \phi_1 \mathbf{G}|| \dots ||\mathbf{A}_\ell + \phi_\ell \mathbf{G}]$$

where $\phi = \mathsf{fsk}||\mathsf{fct}$ and the matrices $\mathbf{A}_i$ live in the public parameters.

Hybrid 1. Generate $\mathsf{fct} = \mathsf{FHE.Enc}(\mathsf{fsk}, f_0)$ as above, and let $\phi = \mathsf{fsk}||\mathsf{fct}$. Choose

$$\mathbf{A}_{f_0} = [\mathbf{A}'_1|| \ldots ||\mathbf{A}'_\ell]$$

to be a truly random LWE matrix of the appropriate dimensions, and program $\mathbf{A}_i$ in the public parameters to be $\mathbf{A}'_i - \phi_i\mathbf{G}$. Hybrid 1 is distributed identically to that in Hybrid 0.

Hybrid 2. Replace $\mathbf{s}\mathbf{A}_{f_0} + \mathbf{e}$ in Hybrid 1 with a uniformly random vector. This is computationally indistinguishable from Hybrid 1 by an application of LWE with respect to the uniformly random matrix $\mathbf{A}_{f_0}$.

Hybrid 3. Replace the public parameters by uniformly random matrices $\mathbf{A}_i$. This hybrid is distributed identically to Hybrid 2. Note that the distribution in this hybrid is independent of the FHE secret key fsk.

Hybrid 4. Replace fct in Hybrid 3 with an encryption of f_1 instead of f_0. This is computationally indistinguishable from Hybrid 3 by an application of FHE semantic security.

The remaining hybrids backtrack through hybrids 2 back to 0 using f_1 instead of f_0.

Hybrid 5–7. This is identical to Hybrid 2–0, except that fct is an encryption of f_1.

Putting the hybrid argument together, we have that given the public parameters pp, the shift keys for f_0 and f_1 are computationally indistinguishable. Indistinguishability relies on LWE for matrices in $\mathbb{Z}_q^{n\times m}$ as well as the semantic security of an FHE scheme (with almost linear decryption) over $\mathbb{Z}_q$ with messages in $\mathbb{Z}_p$, which can both be arranged to follow from the hardness of GapSVP with subexponential approximation factors.

8.4.5 Proof of Sum-Resistance

Assume that an adversary $\mathcal{A}$, given msk and pp, comes up with weights $w_1, \ldots, w_t \in \{-1, 0, 1\}^t \setminus \{0^t\}$ and inputs $x_1, \ldots, x_t$ such that

$$\sum_{i=1}^{t} w_i \cdot \mathsf{Eval}(\mathsf{msk}, x_i) = 0 \pmod{p}$$

and either the x_i are all distinct, or the x_i are not all equal and $\sum_i w_i \neq 0$. The equation above says that

$$\sum_{i=1}^{t} w_i \cdot \lfloor \mathbf{s}\mathbf{A}_{x_i} + \mathbf{u}\mathbf{G}^{-1}(\mathbf{A}_{x_i}) \rceil_p = 0 \pmod{p}$$

Rewriting this, we have

$$\sum_{i=1}^{t} w_i \cdot \lfloor (\mathbf{s}\mathbf{G} + \mathbf{u})\mathbf{G}^{-1}(\mathbf{A}_{x_i}) \rceil_p = \sum_{i=1}^{t} w_i \cdot \left\lfloor \frac{p}{q} \cdot (\mathbf{s}\mathbf{G} + \mathbf{u})\mathbf{G}^{-1}(\mathbf{A}_{x_i}) \right\rceil = 0 \pmod{p}$$

Writing $\mathbf{v}$ for $\mathbf{s}\mathbf{G} + \mathbf{u}$, and isolating the rounding errors $\epsilon_i \in \left(\frac{1}{q}\mathbb{Z}\right)^{\mu}$, we have

$$\frac{p}{q} \cdot \mathbf{v} \cdot \sum_{i=1}^{t} w_i \cdot \mathbf{G}^{-1}(\mathbf{A}_{x_i}) = \sum_{i=1}^{t} w_i \epsilon_i \pmod{p}$$

Note that $\left|\left|\sum_{i=1}^{t} w_i \epsilon_i\right|\right|_{\infty} \leq t$ since $||\epsilon_i||_{\infty} \leq 1$ for all i. Multiplying both sides by q/p,

$$\mathbf{v} \cdot \sum_{i=1}^{t} w_i \cdot \mathbf{G}^{-1}(\mathbf{A}_{x_i}) = \frac{q}{p} \cdot \sum_{i=1}^{t} w_i \epsilon_i := \epsilon \pmod{q}$$

where $\epsilon \in \mathbb{Z}^{\mu}$ and $||\epsilon||_{\infty} \leq qt/p$. Now, we have two possibilities:

Case 1. $\sum_{i=1}^{t} w_i \cdot \mathbf{G}^{-1}(\mathbf{A}_{x_i}) \neq 0 \pmod{q}$. In this case, the matrix $\mathbf{Z} = \sum_{i=1}^{t} w_i \cdot \mathbf{G}^{-1}(\mathbf{A}_{x_i})$ — or any nonzero column $\mathbf{z}$ of $\mathbf{Z}$ — constitutes an approximate $\mathbb{Z}_q$-SIS (Definition 8.19) solution on instance $\mathbf{v}$, with input norm bound $||\mathbf{z}||_{\infty} \leq t$ and output error bound $E = \frac{qt}{p}$.

By Claim 8.20.1, this variant of $\mathbb{Z}_q$-SIS is as hard as approximate $\mathbb{Z}_{\widetilde{q}}$-SIS with the following parameters:

- Modulus $\widetilde{q} = \widetilde{\Theta}(\sqrt{p})$
- Input norm bound $\beta = t$
- Output error bound $\eta = \frac{\widetilde{q}t}{p} + 2mt = \widetilde{O}(\frac{t}{\sqrt{p}}) + 2mt \le 2mt + O(1)$ (since $p = 2^{\Theta(\lambda)}$).

By Fact 8.20, setting $\widetilde{q}$ to be the product of the first $\widetilde{\lambda} \ge \lambda^{1/3}$ primes, this problem is at least as hard as $\mathsf{SIVP}/\mathsf{GapSVP}$ over lattices of dimension $\lambda^{1/3}$ with approximation ratio $\mathsf{poly}(\lambda, m, t)$.

Case 2. $\sum_{i=1}^{t} w_i \cdot \mathbf{G}^{-1}(\mathbf{A}_{x_i}) = 0 \pmod{q}$. In this case, we know that

$$\mathbf{G} \cdot \sum_{i=1}^{t} w_i \cdot \mathbf{G}^{-1}(\mathbf{A}_{x_i}) = \sum_{i=1}^{t} w_i \mathbf{A}_{x_i} = 0 \pmod{q}$$

We now show how to use this to break SIS.

Let $h = h_1 \dots h_\ell$ be the description of a random function chosen from a t-wise independent hash family with range $\mathbb{Z}_q^{\mu}$. Moreover, let $x_1, \dots x_t$ denote the inputs returned by any fixed execution of $\mathcal{A}$. Then, let

$$y = \sum_{i=1}^{t} w_i h(x_i) \pmod{q}.$$

We note that with high probability over the choice of h, we have $y \neq 0$. This follows directly from the t-wise independence of h: if the x_i are distinct, then indeed $\sum_{i=1}^{t} w_i h(x_i)$ is uniformly random (since each $h(x_i)$ is uniform and independent of the other $h(x_j)$). Similarly, if the x_i are not-all-equal and $\sum_i w_i \neq 0$, then there exists a term $\sum_{i \in S} w_i h(x_i)$ corresponding to one "super-variable" where $\sum_{i \in S} w_i \neq 0$, again implying that the overall sum is uniformly random. Therefore, we conclude that with non-negligible probability over the randomness of $\mathcal{A}, \mathsf{msk}, h$, $\mathcal{A}$ outputs $\mathbf{x}$ such that $\sum_{i=1}^{t} w_i \mathbf{G}^{-1}(\mathbf{A}_{x_i}) = 0$ and $y \neq 0$.

Now, imagine the experiment where $\mathbf{A}_j$ is picked as $\mathbf{A}\mathbf{R}_j + h_j\mathbf{G}$. Here,

$$\mathbf{A} = \begin{bmatrix} \mathbf{a} \\ \underline{\mathbf{A}} \end{bmatrix}$$

where $\underline{\mathbf{A}}$ is an SIS challenge matrix and $\mathbf{a}$ is uniformly random. This is statistically indistinguishable from above, so the same claimed property holds. Now,

$$\mathbf{A}_x^{(i)} = \mathsf{Gadget.MEval}(\mathcal{U}_x^{(i)}, \mathbf{A}_1, \ldots, \mathbf{A}_\ell) = \mathbf{A}\mathbf{R}_{x,i} + h(x)|_i\mathbf{G}$$

and

$$\mathsf{a}_x^{(i)} = \mathbf{A}\mathbf{r}_{x,i} + h(x)|_i\mathbf{u}_1$$

where $\mathbf{u}_1$ is the first unit vector. (Technically, $\mathbf{A}_x^{(i)}$ is computed by doing a homomorphic evaluation of h and then decrypting. However, this complication does not make a significant difference to our argument below.)

We know that for each $i \in [\mu]$,

$$\sum_{j=1}^{t} w_j \mathsf{a}_{x_j}^{(i)} = 0 \pmod{q}.$$

Defining $\mathbf{R}_{x_j} = \begin{bmatrix} \mathbf{r}_{x_j,1} & \ldots & \mathbf{r}_{x_j,\mu} \end{bmatrix}$, we conclude that

$$\mathbf{A} \cdot \underbrace{\sum_{j=1}^{t} w_j \mathbf{R}_{x_j}}_{:=\mathbf{R}} + \left[\underbrace{\sum_{j=1}^{t} w_j h_1(x_j)}_{=y_1} \mathbf{u}_1 || \quad \ldots \quad || \underbrace{\sum_{j=1}^{t} w_j h_\mu(x_j)}_{=y_\mu} \mathbf{u}_1 \right] = 0 \pmod{q}$$

Whenever $y \neq 0 \pmod{q}$, it follows that $\mathbf{R}$ is not zero. Now, we have $\underline{\mathbf{A}}\mathbf{R} = 0 \pmod{q}$ (since $\underline{\mathbf{u}_1} = 0$) and $\mathbf{R} \neq 0$ giving us a SIS solution w.r.t. $\underline{\mathbf{A}}$. This finishes the proof of weighted t-sum-resistance.

8.4.6 Putting it Together: Weighted Sum-Resistant SHSFs

Combining the results of Section 8.4.4, Section 8.4.3, and Section 8.4.5, we obtain the following theorem.

Theorem 8.24. *Assume that there is some $\epsilon > 0$ for which it is hard to approximate short vector problems in worst case n-dimensional lattices to within 2^{n^ϵ} factor. Let $\mathsf{SHSF} = (\mathsf{Gen}, \mathsf{Shift}, \mathsf{Eval})$ be the SHSF family constructed above. Then, the hash function family $\mathcal{H}_{\text{plain}}$ that uses $(\mathsf{pp}, \mathsf{msk}) \leftarrow \mathsf{Gen}(1^\lambda)$ as a hash key, and computes the function*

$$h((\mathsf{pp}, \mathsf{msk}), x) = \mathsf{Eval}(\mathsf{pp}, \mathsf{msk}, x)$$

is t-weighted-sum-resistant for every $t = \mathsf{poly}(\lambda)$.

Combining Theorem 8.24 and Theorem 8.23 (the CI lifting theorem), we get a hash family that is CI for shifted (weighted) sum relations.

Theorem 8.25. *Under the same assumption as in Theorem 8.24, there is a hash function family $\mathcal{H}$ that is (R_{out}, f)-correlation intractable (as in Definition 8.14), where R_{out} is the weighted sum relation as in Definition 8.15 and f is any efficiently computable function. That is, $\mathcal{H}$ is correlation-intractable for shifted (weighted) sum relations.*

8.5 Output-Intractable SHSFs from iO

In this section, we present constructions of Output-Intractable SHSFs from iO (Theorem 8.6 and Theorem 8.5). For simplicity, we set the shift modulus $p = 2$ for SHSFs in the remainder of this section.

8.5.1 IO-Related Preliminaries

Indistinguishability Obfuscation

An *obfuscator for all circuits* is a PPT algorithm $\mathcal{O}$ such that for every circuit C, $\mathcal{O}(C)$ is with probability 1 a circuit $\tilde{C}$ with the same functionality as C.

Definition 8.26 (Indistinguishability Obfuscation [BGI+01]). *$\mathcal{O}$ is a (s, δ)-secure* indistinguishability obfuscator *(iO) if for all pairs of functionally equivalent circuits C_0 and C_1 of size $|C_0| = |C_1| = \lambda$, and all circuits $\mathcal{A}$ of size $s(\lambda)$, it holds that*

$$\Pr[\mathcal{A}(\mathcal{O}(C_0)) = 1] - \Pr[\mathcal{A}(\mathcal{O}(C_1)) = 1] \leq O(\delta(\lambda)).$$

Puncturable PRFs

Definition 8.27 (Puncturable PRF [BW13, BGI14, KPTZ13, SW14]). *A puncturable PRF family is a family of functions*

$$\mathcal{F} = \left\{F_{\lambda,s} : \{0,1\}^{\nu(\lambda)} \to \{0,1\}^{\mu(\lambda)}\right\}_{\lambda \in \mathbb{N}, s \in \{0,1\}^{\ell(\lambda)}}$$

with associated (deterministic) polynomial-time algorithms $(\mathcal{F}.\mathsf{Eval}, \mathcal{F}.\mathsf{Puncture}, \mathcal{F}.\mathsf{PuncEval})$ satisfying

- *For all $x \in \{0,1\}^{\nu(\lambda)}$ and all $s \in \{0,1\}^{\ell(\lambda)}$, $\mathcal{F}.\mathsf{Eval}(s, x) = F_{\lambda,s}(x)$.*
- *For all distinct $x, x' \in \{0,1\}^{\nu(\lambda)}$ and all $s \in \{0,1\}^{\ell(\lambda)}$,*

$$\mathcal{F}.\mathsf{PuncEval}(\mathcal{F}.\mathsf{Puncture}(s, x), x') = \mathcal{F}.\mathsf{Eval}(s, x')$$

For ease of notation, we write $F_s(x)$ and $\mathcal{F}.\mathsf{Eval}(s, x)$ interchangeably, and we write $s\{x\}$ to denote $\mathcal{F}.\mathsf{Puncture}(s, x)$.

$\mathcal{F}$ is said to be (s, δ)-secure if for every $\{x^{(\lambda)} \in \{0,1\}^{\nu(\lambda)}\}_{\lambda \in \mathbb{N}}$, the following two distribution ensembles (indexed by λ) are $\delta(\lambda)$-indistinguishable to circuits of size $s(\lambda)$:

$$(S\{x^{(\lambda)}\}, F_S(x^{(\lambda)})) \text{ where } S \leftarrow \{0,1\}^{\ell(\lambda)}$$

and

$$(S\{x^{(\lambda)}\}, U) \text{ where } S \leftarrow \{0,1\}^{\ell(\lambda)}, U \leftarrow \{0,1\}^{\mu(\lambda)}.$$

Theorem 8.28 ([GGM84, KPTZ13, BW13, BGI14, SW14]). *If {polynomially secure, subexponentially secure} one-way functions exist, then for all functions μ :*

$\mathbb{N} \to \mathbb{N}$ *(with* $1^{\mu(\nu)}$ *polynomial-time computable from* 1^ν*), and all* $\delta : \mathbb{N} \to [0,1]$ *with* $\delta(\nu) \geq 2^{-\mathsf{poly}(\nu)}$*, there are polynomials* $\ell(\lambda), \nu(\lambda)$ *and a {polynomially secure,* $(\frac{1}{\delta(\nu(\lambda))}, \delta(\nu(\lambda)))$*-secure} puncturable PRF family*

$$\mathcal{F}_\mu = \left\{ F_{\lambda,s} : \{0,1\}^{\nu(\lambda)} \to \{0,1\}^{\mu(\nu(\lambda))} \}_{\lambda \in \mathbb{N}, s \in \{0,1\}^{\ell(\lambda)}} \right\}.$$

Lossy Functions

Definition 8.29 (Lossy Functions [PW08]). *A* lossy function family $\mathsf{LF} = (\mathsf{LF.Gen}, \mathsf{LF.Eval})$ *consists of two PPT algorithms:*

- $\mathsf{LF.Gen}(1^\lambda, \text{injective/lossy})$ *outputs an evaluation key* ek *either in "injective mode" or "lossy mode."*

- $\mathsf{LF.Eval}(\mathsf{ek}, x)$ *takes an evaluation key* ek *as well as an input* $x \in \{0,1\}^{\nu(\lambda)}$*. It returns a* deterministic *output* $y \in \{0,1\}^{N(\lambda)}$.

We require that LF *satisfies three properties:*

- ***Injectivity****: With probability* $1 - \text{negl}(\lambda)$ *over the randomness of* $\mathsf{ek} \leftarrow \mathsf{LF.Gen}(1^\lambda, \text{injective})$*, the function* $\mathsf{LF.Eval}(\mathsf{ek}, \cdot)$ *is injective.*

- ***Lossiness*** *(with parameter* $\ell(\lambda)$*): With probability* $1 - \text{negl}(\lambda)$ *over the randomness of* $\mathsf{ek} \leftarrow \mathsf{LF.Gen}(1^\lambda, \text{lossy})$*, the range of the function* $\mathsf{LF.Eval}(\mathsf{ek}, \cdot)$ *has size at most* $2^{\ell(\lambda)}$.

- ***Key Indistinguishability****: randomly sampled injective and lossy keys are computationally indistinguishable.*

8.5.2 Output-Intractable SHSFs from iO + Output-Intractable Puncturable PRFs

In this section, we note that the natural construction of SHSFs from (subexponential) iO and puncturable PRFs (following the [BLW17] construction of private constrained PRFs from iO) also yields output-intractable SHSFs from iO along with output-intractable puncturable PRFs. This fact will be used in all of our iO-based constructions.

Construction 8.30 (SHSF from IO)**.** *Let* $\mathsf{PRF} = \{F_s : \{0,1\}^{\nu(\lambda)} \to \{0,1\}^{\mu(\lambda)}\}$ *denote a (puncturable) PRF family and let* $\mathcal{O}$ *denote an indistinguishability obfuscator. Then,* PRF *can be augmented with the algorithm* Shift*, defined as follows:*

$$\mathsf{Shift}(s, f) = \mathcal{O}\Big(x \mapsto \mathsf{PRF}_s(x) + f(x)\Big).$$

Moreover, a shifted key $\mathsf{sk}_f \leftarrow \mathsf{Shift}(s, f)$ *can be evaluated on an input* x *simply by interpreting* sk_f *as a program and evaluating* $\mathsf{sk}_f(x)$*.*

Lemma 8.31. *Suppose that* PRF *is a* $2^{-\nu(\lambda)} \cdot \mathrm{negl}(\lambda)$*-secure puncturable PRF (Definition 8.27), and* $\mathcal{O}$ *is a* $2^{-\nu(\lambda)} \cdot \mathrm{negl}(\lambda)$ *secure indistinguishability obfuscator (Definition 8.26).*

Then, $(\mathsf{PRF}, \mathsf{Shift})$ *is a SHSF for bounded-size shift functions. Moreover, if the hash family* $\mathcal{H}_{\mathrm{plain}}(\mathsf{msk}, x) = \mathsf{PRF}_{\mathsf{msk}}(x)$ *is output-intractable (or NAE-output-intractable) for a relation* R_{out}*, then the same is true for* SHSF*.*

Proof. For the first claim, it suffices to show that $(\mathsf{PRF}, \mathsf{Shift})$ satisfies correctness and shift-hiding. Correctness follows immediately from the correctness of $\mathcal{O}$.

To see that $(\mathsf{PRF}, \mathsf{Shift})$ is shift-hiding – namely, that $\mathsf{sk}_f \approx_c \mathsf{sk}_g$ for any pair of (bounded-size) circuits (f, g), we closely follows the CHCPRF security proof in [BLW17]. Namely, we appeal to a hybrid argument with $2^\nu + 2$ hybrid distributions on keys sk, defined as follows:

- Hyb_{-1}: $\mathsf{sk} = \mathsf{sk}_f \leftarrow \mathsf{Shift}(s, f) = \mathcal{O}\left(x \mapsto \mathsf{PRF}_s(x) + f(x)\right)$.
- For every $0 \le x^* \le 2^\nu - 1$ (interpreting x^* as both an integer and a string $\mathsf{Hyb}_{x^*} = \mathsf{sk} \leftarrow \mathcal{O}\left(x \mapsto \mathsf{PRF}_s(x) + g(x) \text{ if } x < x^*, x \mapsto \mathsf{PRF}_s(x) + f(x) \text{ if } x \ge x^*\right)$
- Hyb_{2^ν}: $\mathsf{sk} = \mathsf{sk}_g \leftarrow \mathsf{Shift}(s, g) = \mathcal{O}\left(x \mapsto \mathsf{PRF}_s(x) + g(x)\right)$.

We note that $\mathsf{Hyb}_{-1} \approx_{c,2^{-\nu}\mathrm{negl}(\lambda)} \mathsf{Hyb}_0$ and $\mathsf{Hyb}_{2^\nu-1} \approx_{c,2^{-\nu}\mathrm{negl}(\lambda)} \mathsf{Hyb}_{2^\nu}$ by the $2^{-\nu} \cdot \mathrm{negl}(\lambda)$-security of $\mathcal{O}$. Additionally, we note that $\mathsf{Hyb}_{x^*} \approx_{c,O(2^{-\nu}\cdot\mathrm{negl}(\lambda))} \mathsf{Hyb}_{x^*+1}$ for every $0 \leq x^* \leq 2^\nu - 2$ by a standard puncturing argument. This relies on the $2^{-\nu} \cdot \mathrm{negl}(\lambda)$-security of both the obfuscator and the puncturable PRF. This completes the proof of shift-hiding.

Finally, since the honest evaluation of the SHSF in Construction 8.30 is identical to a puncturable PRF evaluation (with the same secret key), we note that the SHSF SHSF is (NAE) output-intractable for a relation R_{out} if and only if PRF is (NAE) output-intractable for the same relation R_{out}. Thus, by Theorem 8.23, in order to obtain correlation-intractable hash functions based on IO, we have reduced the problem to constructing output-intractable $2^{-\nu}$-secure puncturable PRFs.

□

We now present two constructions of $2^{-\nu}$-secure puncturable PRFs, based on different assumptions.

8.5.3 Construction 1: Postcomposition with an Output-Intractable Hash

Construction 8.32. *Let* PRF *denote a puncturable PRF family mapping* $\{0,1\}^{\nu(\lambda)} \to \{0,1\}^{N(\lambda)}$*. Let* $\mathcal{H}$ *denote an* R_{out}*-output intractable hash family mapping* $\{0,1\}^{N(\lambda)} \to \{0,1\}^{\mu(\lambda)}$*. Then, we define the PRF family* $\mathsf{PRF}_{\mathcal{H}} = \mathcal{H} \circ \mathsf{PRF}$ *as follows:*

- *A secret key for* $\mathsf{PRF}_{\mathcal{H}}$ *is a pair* (k, sk) *with* $k \leftarrow \mathcal{H}.\mathsf{Gen}(1^\lambda)$ *and* $\mathsf{sk} \leftarrow \mathsf{PRF}.\mathsf{Gen}(1^\lambda)$.
- *Evaluation is defined to be*

$$\mathsf{PRF}_{\mathcal{H}}(k, \mathsf{sk}, x) = h(k, \mathsf{PRF}_{\mathsf{sk}}(x)).$$

Lemma 8.33. *Suppose that* PRF *is a* $2^{-\nu} \cdot \mathrm{negl}(\lambda)$*-secure puncturable PRF family that is* injective *with high probability,* $\mathcal{H}$ *is* R_{out}*-output intractable (or NAE-*R_{out}*-output intractable), and* $\mathcal{H}$ *has a* nearly uniform output distribution*, meaning that*

$$\left\{k \leftarrow \mathcal{H}.\mathsf{Gen}(1^\lambda), x \leftarrow \{0,1\}^{N(\lambda)} : (k, h(x))\right\}$$

$$\approx_{c,2^{-\nu}\cdot\mathrm{negl}(\lambda)} \left\{k \leftarrow \mathcal{H}.\mathsf{Gen}(1^\lambda), y \leftarrow \{0,1\}^{\mu(\lambda)} : (k,y)\right\}.$$

Then, $\mathsf{PRF}_{\mathcal{H}}$ *is a* $2^{-\nu}\cdot\mathrm{negl}(\lambda)$*-secure puncturable PRF family that is also* R_{out}*-output intractable (or NAE-*R_{out}*-output intractable).*

Proof. We first show output intractability. If an adversary $\mathcal{A}(k, \mathsf{sk})$ finds distinct (respectively, not-all-equal) inputs $(x_1, \ldots, x_t)$ such that $(h_k(\mathsf{PRF}_{\mathsf{sk}}(x_1), \ldots, h_k(\mathsf{PRF}_{\mathsf{sk}}(x_t))) \in R_{\mathrm{out}}$ with non-negligible probability, then we claim that this violates the R_{out}-output intractability of $\mathcal{H}$. This holds because with all but negligible probability, $\mathsf{PRF}_{\mathsf{sk}}$ is an injective function, in which case the inputs $\mathsf{PRF}_{\mathsf{sk}}(x_1), \ldots, \mathsf{PRF}_{\mathsf{sk}}(x_t)$ to h_k are distinct (respectively, not-all-equal) as long as $x_1, \ldots, x_t$ are distinct (respectively, not-all-equal). This gives an attack on the R_{out}-output intractability of $\mathcal{H}$: given a key k, an adversary $\mathcal{A}'$ can sample sk, call $(x_1, \ldots, x_t) \leftarrow \mathcal{A}(k, \mathsf{sk})$, and output $(\mathsf{PRF}_{\mathsf{sk}}(x_1), \ldots, \mathsf{PRF}_{\mathsf{sk}}(x_t))$.

Next, we show that $\mathsf{PRF}_{\mathcal{H}}$ is a $2^{-\nu}\mathrm{negl}(\lambda)$-secure puncturable PRF family. To do so, we define a puncturing algorithm:

$$\mathsf{PRF}_{\mathcal{H}}.\mathsf{Puncture}(k, \mathsf{sk}, x^*) = (k, \mathsf{sk}\{x^*\}).$$

One can then verify that for $x \neq x^*$

$$\mathsf{PuncEval}((k, \mathsf{sk})\{x^*\}, x) = \mathsf{PRF}_{\mathcal{H}}(k, \mathsf{sk}, x).$$

Finally, $2^{-\nu}\cdot\mathrm{negl}(\lambda)$-pseudorandomness at punctured points follows from the analogous property for PRF along with the fact that $\mathcal{H}$ has a nearly uniform output distribution.

□

8.5.4 Construction 2: Precomposition with a Lossy Function

Construction 8.34. *Let* PRF *denote a puncturable PRF family mapping* $\{0,1\}^{N(\lambda)} \to \{0,1\}^{\mu(\lambda)}$*. Let* $\mathsf{LF} = (\mathsf{LF.Gen}, \mathsf{LF.Eval})$ *denote a lossy function family mapping* $\{0,1\}^{\nu(\lambda)} \to \{0,1\}^{N(\lambda)}$ *and lossiness parameter* $\ell(\lambda)$*. Then, we define the PRF family* $\mathsf{PRF}_{\mathsf{LF}} =$

$\mathsf{PRF} \circ \mathsf{LF}$ *as follows:*

- *A secret key for* $\mathsf{PRF}_{\mathsf{LF}}$ *is a pair* $(\mathsf{sk}, \mathsf{ek})$ *with* $\mathsf{ek} \leftarrow \mathsf{LF.Gen}(1^\lambda, \text{injective})$ *and* $\mathsf{sk} \leftarrow \mathsf{PRF.Gen}(1^\lambda)$.
- *Evaluation is defined to be*

$$\mathsf{PRF}_{\mathsf{LF}}(\mathsf{sk}, \mathsf{ek}, x) = \mathsf{PRF}(\mathsf{sk}, \mathsf{LF.Eval}(\mathsf{ek}, x)).$$

Lemma 8.35. *Suppose that* PRF *is a* $\left(2^{N(\lambda)+\ell(\lambda)t(\lambda)}, 2^{-\nu(\lambda)} \cdot \mathrm{negl}(\lambda)\right)$*-secure puncturable PRF family, and suppose that* LF *is a lossy function family with lossiness parameter* $\tau(\lambda)$.
Then, for any relation R_{out} *with sparsity at most* $2^{-t(\lambda)\ell(\lambda)} \cdot \mathrm{negl}(\lambda)$, $\mathsf{PRF}_{\mathsf{LF}}$ *is a* $2^{-\nu} \cdot \mathrm{negl}(\lambda)$*-secure puncturable PRF family that is also* R_{out}*-output intractable.*

Moreover, if R_{out} *is also sparse whenever the inputs* $x_1, \ldots, x_t$ *are not-all-equal, then the PRF family satisfies NAE-*R_{out}*-output intractability.*

Proof. We first show puncturing-pseudorandomness. To do so, we define a puncturing algorithm

$$\mathsf{PRF}_{\mathcal{H}}.\mathsf{Puncture}(\mathsf{sk}, \mathsf{ek}, x^*) = (k, \mathsf{sk}\{\mathsf{LF.Eval}(x^*)\}).$$

Punctured evaluation correctness (with all but negligible probability over the sampling of $(\mathsf{sk}, \mathsf{ek})$) follows from the fact that ek is sampled in injective mode. Pseudorandomness follows directly from the pseudorandomness of PRF.

We next show output intractability. If an adversary $\mathcal{A}(\mathsf{sk}, \mathsf{ek})$ finds distinct (respectively, not-all-equal) inputs $(x_1, \ldots, x_t)$ such that

$$(\mathsf{PRF}_{\mathsf{sk}}(\mathsf{LF.Eval}(\mathsf{ek}, x_1)), \ldots, \mathsf{PRF}_{\mathsf{sk}}(\mathsf{LF.Eval}(\mathsf{ek}, x_t))) \in R_{\mathrm{out}}$$

with non-negligible probability ϵ, then since ek is sampled in injective mode, the same claim holds where $(\mathsf{LF.Eval}(\mathsf{ek}, x_1), \ldots, \mathsf{LF.Eval}(\mathsf{ek}, x_t))$ are distinct (respectively, not-all-equal).

Then, by the security of LF, we also know that when $\mathsf{ek} \leftarrow \mathsf{LF.Gen}(1^\lambda, \text{lossy})$ is sampled from the *lossy* distribution, we have that

$$(x_1, \dots, x_t) \leftarrow \mathcal{A}(\mathsf{sk}, \mathsf{ek}) : (\mathsf{LF.Eval}(\mathsf{ek}, x_1), \dots, \mathsf{LF.Eval}(\mathsf{ek}, x_t)) \text{ are distinct}$$

$$\text{and } (\mathsf{PRF}_{\mathsf{LF}}(\mathsf{sk}, \mathsf{ek}, x_1), \dots, \mathsf{PRF}_{\mathsf{LF}}(\mathsf{sk}, \mathsf{ek}, x_t)) \in R_{\text{out}} \geq \epsilon - \text{negl}(\lambda).$$

Finally, we claim that in reality, with high probability over $(\mathsf{sk}, \mathsf{ek})$, *there do not exist such input tuples.* This follows from the pseudorandomness of PRF: for any fixed set S of size $2^{\ell(\lambda)}$, the probability that a random function F has an t-tuple of distinct (respectively, not-all-equal) inputs $z_1, \dots, z_t$ from S such that $(F(z_1), \dots, F(z_t)) \in R_{\text{out}}$ is at most $|S|^t \cdot \beta$ if R_{out} has sparsity β, which is negligible under our hypotheses. Picking $S = \text{Im}(\mathsf{LF}(\mathsf{ek}, \cdot))$, we conclude that the same holds for the PRF family $\mathsf{PRF}_{\mathsf{sk}}$, as this condition can be tested in time $2^{N(\lambda)+\ell(\lambda)t(\lambda)}$ by enumeration. Thus, we obtain a contradiction, completing the proof of Lemma 8.35.

□

8.5.5 Putting it Together

Combining Theorem 8.23 and Lemma 8.31 with Lemma 8.33 and Lemma 8.35, respectively, we obtain our final constructions of correlation intractable hash families based on obfuscation. We restate the results (Theorem 8.6 and Theorem 8.5) from the introduction for completeness.

Theorem 8.36 (Theorem 8.6, restated)**.** *Assume the existence of*

1. *Subexponentially secure indistinguishability obfuscation,*

2. *Subexponentially secure one-way functions, and*

3. *A hash family $\mathcal{H}$ such that (i) $\mathcal{H}$ is R_{out}-output intractable, and (ii) for a random input X, $h_k(X)$ is $2^{-\nu} \cdot \text{negl}(\lambda)$-indistinguishable from uniform (even given k).*

Then, there exists a hash family that is CI for shifted R_{out}-relations.

This follows by combining Theorem 8.23, Lemma 8.31, and Lemma 8.33.

Theorem 8.37 (Theorem 8.5, restated)**.** *Assume the existence of*

1. *Subexponential IO,*
2. *Subexponential OWFs, and*
3. *Lossy functions with input domain $\{0,1\}^\nu$ with a range of size $\leq 2^\ell$ in lossy mode.*

Then, there exists a hash family $\mathcal{H}$ that is CI for all (efficiently decidable) shifted t-ary output relations with sparsity at most $2^{-t\ell}$.

This follows by combining Theorem 8.23, Lemma 8.31, and Lemma 8.35.

Appendix A

Fiat-Shamir from CI, without using a Commitment Trapdoor

In this brief chapter, which is appendix from [CLMQ21], we address the following nagging issue about all (other) Fiat-Shamir instantiations in this thesis. Consider using a hash function family $\mathcal{H}$ to instantiate the Fiat-Shamir heuristic for a protocol Π. If we want to prove security of $\Pi_{\mathrm{FS},\mathcal{H}}$ solely based on the fact that $\mathcal{H}$ is correlation-intractable for *efficiently computable* functions, we have needed the *interactive* protocol Π to have some "trapdoor" enabling an efficient algorithm for computing bad challenges for Π. For example, if Π is Blum's Hamiltonicity protocol [Blu86], then we needed to instantiate the (generic) commitment scheme as an *extractable* commitment scheme (e.g. using public-key encryption). While this is a valid and interesting instantiation, it does not address what happens for *other* choices of commitment scheme.

Are there ways to instantiate the commitment scheme *without* resorting to public-key cryptography, while still allowing for a Fiat-Shamir instantiation from standard assumptions? In this section, we note one such way: one can also use a *random-oracle based* commitment! Of course, the whole point of this thesis is to use *concrete* cryptographic primitives and not resort to heuristic models. Nevertheless, we leave this short result as a philosophical point that, in contrast to "trapdoored" commitment schemes that have been used so far, Blum's protocol using a "maximally unstructured" commitment scheme also admits standard-model Fiat-Shamir hash functions.

Specifically, in this section, we show that correlation intractability for efficiently computable functions [CCH$^+$19, PS19] implies a sound instantiation of Fiat-Shamir for the following idealized variant of the Blum Hamiltonicity protocol [Blu86].

$P(G,\sigma)$		$V(G)$
$\pi \leftarrow S_n$, $G' = \pi(G)$ $\alpha \leftarrow \mathsf{Com}(G' \Vert \pi)$	$\xrightarrow{\alpha}$	
	$\xleftarrow{\beta}$	$\beta \leftarrow \{0,1\}$
If $\beta = 0$, decommit to (G', π). If $\beta = 1$, reveal $\pi \circ \sigma$ and decommit to the edges in G' corresponding to the cycle $\pi \circ \sigma$.	$\xrightarrow{\gamma}$	Accept if all decommitments are correct and: either $\beta = 0$ and $G' = \pi(G)$ or $\beta = 1$ and all edge decommitments are 1.

Figure A-1: A Modified Idealized Blum Protocol Π

As is typical for these results, we require the prover to additionally commit to the permutation π and decommit to π if $\beta = 0$. In this case, the verifier checks that π is a valid permutation and that $G' = \pi(G)$. The reason this modification is made is so that given a (partial) decommitment to the first message α, it is possible to efficiently decide which challenge is answerable using this decommitment. In the original Blum protocol, the analogous computation requires solving a graph isomorphism problem.

Next, we instantiate $\mathsf{Com}(b; r) = \mathcal{O}(b, r)$ using a random oracle. Concretely, we set $|r| = \lambda = \lambda(n)$ and $|\mathcal{O}(b, r)| = \kappa = \kappa(n)$ to be arbitrary polynomial functions in $n = |V(G)|$. The protocol above is then repeated $t = t(n)$ times in parallel to obtain negligible soundness error. We then prove:

Theorem A.1. *Suppose that for every (efficiently computable) $s(n) = \mathsf{poly}(n)$, there exists a hash family $\mathcal{H} = \{h_k : \{0,1\}^{m(n)\kappa(n)t(n)} \to \{0,1\}^{t(n)}\}_{k \in \{0,1\}^{\ell(n)}}$ (for $m(n) = n^2 + n$) that is correlation intractable for all functions computable by size $s(n)$ circuits.*

Then, for an appropriate fixed choice of function $s(\cdot)$, the same hash family $\mathcal{H}$ soundly instantiates the Fiat-Shamir heuristic for the protocol Π^t in the random oracle model.

Proof. Let $\mathcal{H}$ be a family of correlation-intractable hash functions with parameters as above (for $s = s(n)$ chosen appropriately large). Since correlation-intractable hash functions imply the existence of one-way functions, we additionally let $F_s : \{0,1\}^{\kappa(n)-1} \to \{0,1\}$ be a PRF family computable by a family of circuits of size $s(n)$.

Now, suppose that an efficient adversary $\mathcal{A}^{\mathcal{O}(\cdot)}$, given a non-Hamiltonian graph G and random hash function h, breaks the soundness of $\Pi^t_{\mathrm{FS},\mathcal{H}}$ on G.

Let $\tau = \tau(\mathcal{A}, \mathcal{O})$ denote the transcript of $\mathcal{O}$-queries made by $\mathcal{A}$; that is, for every i, $\tau_i = (b_i, r_i, c_i)$ where (b_i, r_i) is the ith query made by $\mathcal{A}$ to $\mathcal{O}$, and $c_i = \mathcal{O}(b_i, r_i)$. Finally, let $(\alpha^*, \beta^*, \gamma^*)$ denote the output of $\mathcal{A}$.

Given an arbitrary first message α and transcript τ, we say that a challenge β is a **bad challenge** for (α, τ) if the following conditions hold:

- For every i such that $\beta_i = 0$, the string of commitments $\alpha_i = (c_{i,0}, \ldots, c_{i,m})$ is entirely contained within the transcript τ, and the corresponding bits $\{b_{i,j}\}$ consist of a permutation π and the graph $\pi(G)$.

- For every i such that $\beta_i = 1$, the transcript τ contains a substring of α_i consisting of commitments to a cycle.

We now note a sequence of facts about the execution of $\mathcal{A}$.

Claim A.1.1. *The probability that $\mathcal{A}^{\mathcal{O}(\cdot)}$ wins with output $(\alpha^*, \beta^*, \gamma^*)$* and *$\beta^*$ is* not *a bad challenge for (α^*, τ) is negligible.*

This claim follows from binding properties of the (random oracle) commitment scheme. This is because if β^* is not bad for (α^*, τ) but $(\alpha^*, \beta^*, \gamma^*)$ is accepting, then γ^* contains decommitments to bits that are not present in τ; this means that $\mathcal{A}^{\mathcal{O}(\cdot)}$ solves an (unconditionally) hard problem in the random oracle model.

Claim A.1.2. *The probability that (α^*, τ) has* multiple *bad challenges associated to it is negligible.*

This again follows from binding properties of the commitment scheme, and the fact that G is not Hamiltonian. Since G is Hamiltonian, if no string c appears twice (for two different choices of (b, r)) in the transcript τ, bad challenges for any (α, τ)

are unique (as each α_i cannot have an opening to both a permutation of G and a Hamiltonian graph simultaneously). However, τ only contains the same commitment string c twice with negligible probability, since it is (unconditionally) hard to find $\mathcal{O}$-collisions.

Thus, given a transcript τ and message α, we define the efficiently computable "transcript bad-challenge function" $f(\tau, \alpha)$ as follows:

- If α_i is present in τ as a commitment to (G', π) and $G' = \pi(G)$, set $\beta_i = 0$.
- Otherwise, set $\beta_i = 1$.
- Output $\beta = (\beta_1, \ldots, \beta_t)$.

By the above analysis, we conclude:

Claim A.1.3. *With non-negligible probability, the adversary $\mathcal{A}^{\mathcal{O}}(G, h)$ outputs $(\alpha^*, \beta^*, \gamma^*)$ such that*

- $\beta^* = h(\alpha^*) = f(\alpha^*, \tau)$, *and*
- τ *contains all necessary decommitments to answer the challenge β^*.*

Note that Claim A.1.3 is an efficiently decidable property of $(\tau, \alpha^*, \beta^*)$. Thus, Claim A.1.3 *also* holds if we replace the truly random oracle $\mathcal{O}$ with the following oracle distribution $\mathcal{O}'$:

- $\mathcal{O}'$ has a hard-coded random seed s for the PRF $F_s : \{0,1\}^{\kappa(n)-1} \to \{0,1\}$
- $\mathcal{O}'(b, r)$ samples a uniformly random $r' \leftarrow \{0,1\}^{\kappa(n)-1}$ and outputs $(r', F_s(r') \oplus b)$.

This follows directly from the pseudorandomness property of the PRF family. Finally, we define the following efficiently computable function $g_s : \{0,1\}^{m(n)\kappa(n)t(n)} \to \{0,1\}^{t(n)}$.

- Input: $\alpha = (\alpha_1, \ldots, \alpha_n)$

- For all i, let $\alpha_i = (c_{i,1}, \ldots, c_{i,m(n)})$ and $c_{i,j} = r'_{i,j}||b'_{i,j}$. Compute $b_{i,j} = F_s(r'_{i,j}) \oplus b'_{i,j}$.
- Let $\tilde{\tau}$ denote a transcript containing triples of the form $(b_{i,j}, r_{i,j}, c_{i,j})$ where $r_{i,j}$ are arbitrary.
- Output $f(\alpha, \tilde{\tau})$.

We claim that $\mathcal{A}^{\mathcal{O}'(\cdot)}$ breaks the correlation intractability of $\mathcal{H}$ with respect to the function g_s. Indeed, whenever the conditions of Claim A.1.3 hold, we also claim that $h(\alpha^*) = g_s(\alpha^*)$. To see this, we note that any commitment $c = (r', b')$ occurring as (b, r, c) in the transcript τ must satisfy the property $b' = F_s(r') \oplus b$. Thus, the ith bit $f(\alpha^*, \tau)_i = 0$ if and only if the ith bit $g_s(\alpha^*)_i = 0$.

We conclude that $\mathcal{A}^{\mathcal{O}'(\cdot)}$, which can be implemented efficiently given the PRF seed s, contradicts the correlation intractability of $\mathcal{H}$ with respect to g_s. Therefore, the protocol $\Pi^t_{\mathrm{FS},\mathcal{H}}$ is indeed sound in the ROM. □

Bibliography

[AABN02] Michel Abdalla, Jee Hea An, Mihir Bellare, and Chanathip Namprempre. From identification to signatures via the Fiat-Shamir transform: Minimizing assumptions for security and forward-security. In Lars R. Knudsen, editor, *EUROCRYPT 2002*, volume 2332 of *LNCS*, pages 418–433. Springer, Heidelberg, April / May 2002.

[ABF$^+$20] Martin Albrecht, Shi Bai, Pierre-Alain Fouque, Paul Kirchner, Damien Stehlé, and Weiqiang Wen. Faster enumeration-based lattice reduction: Root hermite factor $k^{1/(2k)}$ in time $k^{k/8+o(k)}$. In *CRYPTO*, 2020.

[ACPS09] Benny Applebaum, David Cash, Chris Peikert, and Amit Sahai. Fast cryptographic primitives and circular-secure encryption based on hard learning problems. In *Advances in Cryptology-CRYPTO 2009*, pages 595–618. Springer, 2009.

[AD97] Miklós Ajtai and Cynthia Dwork. A public-key cryptosystem with worst-case/average-case equivalence. In *Proceedings of the twenty-ninth annual ACM symposium on Theory of computing*, pages 284–293. ACM, 1997.

[Adl79] Leonard Adleman. A subexponential algorithm for the discrete logarithm problem with applications to cryptography. In *20th Annual Symposium on Foundations of Computer Science (SFCS 1979)*, pages 55–60. IEEE, 1979.

[ADRS15] Divesh Aggarwal, Daniel Dadush, Oded Regev, and Noah Stephens-Davidowitz. Solving the shortest vector problem in 2n time using discrete gaussian sampling. In *STOC 2015*, pages 733–742, 2015.

[AFV11] Shweta Agrawal, David Mandell Freeman, and Vinod Vaikuntanathan. Functional encryption for inner product predicates from learning with errors. In Dong Hoon Lee and Xiaoyun Wang, editors, *ASIACRYPT 2011*, volume 7073 of *LNCS*, pages 21–40. Springer, Heidelberg, December 2011.

[AG11] Sanjeev Arora and Rong Ge. New algorithms for learning in presence of errors. In *Automata, Languages and Programming - 38th International Colloquium, ICALP 2011, Zurich, Switzerland, July 4-8, 2011, Proceedings, Part I*, pages 403–415, 2011.

[AIK04] Benny Applebaum, Yuval Ishai, and Eyal Kushilevitz. Cryptography in nc0. In *Foundations of Computer Science, 2004. Proceedings. 45th Annual IEEE Symposium on*, pages 166–175. IEEE, 2004.

[AIK11] Benny Applebaum, Yuval Ishai, and Eyal Kushilevitz. How to garble arithmetic circuits. In *Proceedings of the 2011 IEEE 52nd Annual Symposium on Foundations of Computer Science*, pages 120–129. IEEE Computer Society, 2011.

[AJL$^+$19] Prabhanjan Ananth, Aayush Jain, Huijia Lin, Christian Matt, and Amit Sahai. Indistinguishability obfuscation without multilinear maps: New paradigms via low degree weak pseudorandomness and security amplification. In *CRYPTO*, 2019.

[Ajt96] Miklós Ajtai. Generating hard instances of lattice problems (extended abstract). In *28th ACM STOC*, pages 99–108. ACM Press, May 1996.

[AK97] Vikraman Arvind and Johannes Köbler. On resource-bounded measure and pseudorandomness. In *International Conference on Foundations of Software Technology and Theoretical Computer Science*, pages 235–249. Springer, 1997.

[AKS01] Miklós Ajtai, Ravi Kumar, and Dandapani Sivakumar. A sieve algorithm for the shortest lattice vector problem. In *Proceedings of the thirty-third annual ACM symposium on Theory of computing*, pages 601–610. ACM, 2001.

[AKV04] Tim Abbot, Daniel Kane, and Paul Valiant. On algorithms for nash equilibria. *Unpublished manuscript*, page 1, 2004.

[ALM$^+$92] Sanjeev Arora, Carsten Lund, Rajeev Motwani, Madhu Sudan, and Mario Szegedy. Proof verification and hardness of approximation problems. In *33rd FOCS*, pages 14–23. IEEE Computer Society Press, October 1992.

[AP14] Jacob Alperin-Sheriff and Chris Peikert. Faster bootstrapping with polynomial error. In Juan A. Garay and Rosario Gennaro, editors, *CRYPTO 2014, Part I*, volume 8616 of *LNCS*, pages 297–314. Springer, Heidelberg, August 2014.

[App11] Benny Applebaum. Key-dependent message security: Generic amplification and completeness. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT)*, pages 527–546. Springer, 2011.

[AS15] Gilad Asharov and Gil Segev. Limits on the power of indistinguishability obfuscation and functional encryption. In Venkatesan Guruswami, editor, *56th FOCS*, pages 191–209. IEEE Computer Society Press, October 2015.

[Bar01] Boaz Barak. How to go beyond the black-box simulation barrier. In *42nd FOCS*, pages 106–115. IEEE Computer Society Press, October 2001.

[BBBF18] Dan Boneh, Joseph Bonneau, Benedikt Bünz, and Ben Fisch. Verifiable delay functions. In *Annual International Cryptology Conference (EUROCRYPT 2018)*, pages 757–788. Springer, 2018.

[BBC$^+$14] Boaz Barak, Nir Bitansky, Ran Canetti, Yael Tauman Kalai, Omer Paneth, and Amit Sahai. Obfuscation for evasive functions. In Yehuda Lindell, editor, *TCC 2014*, volume 8349 of *LNCS*, pages 26–51. Springer, Heidelberg, February 2014.

[BBH$^+$19] James Bartusek, Liron Bronfman, Justin Holmgren, Fermi Ma, and Ron D. Rothblum. On the (in)security of kilian-based SNARGs. In Dennis Hofheinz and Alon Rosen, editors, *TCC 2019, Part II*, volume 11892 of *LNCS*, pages 522–551. Springer, Heidelberg, December 2019.

[BCCT13] Nir Bitansky, Ran Canetti, Alessandro Chiesa, and Eran Tromer. Recursive composition and bootstrapping for snarks and proof-carrying data. In *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*, pages 111–120. ACM, 2013.

[BCKP14] Nir Bitansky, Ran Canetti, Yael Tauman Kalai, and Omer Paneth. On virtual grey box obfuscation for general circuits. In *International Cryptology Conference*, pages 108–125. Springer, 2014.

[BCM$^+$18] Zvika Brakerski, Paul Christiano, Urmila Mahadev, Umesh V. Vazirani, and Thomas Vidick. A cryptographic test of quantumness and certifiable randomness from a single quantum device. In Mikkel Thorup, editor, *59th FOCS*, pages 320–331. IEEE Computer Society Press, October 2018.

[BCS16] Eli Ben-Sasson, Alessandro Chiesa, and Nicholas Spooner. Interactive oracle proofs. In Martin Hirt and Adam D. Smith, editors, *TCC 2016-B, Part II*, volume 9986 of *LNCS*, pages 31–60. Springer, Heidelberg, October / November 2016.

[BDG$^+$13] Nir Bitansky, Dana Dachman-Soled, Sanjam Garg, Abhishek Jain, Yael Tauman Kalai, Adriana López-Alt, and Daniel Wichs. Why "Fiat-Shamir for proofs" lacks a proof. In Amit Sahai, editor, *TCC 2013*, volume 7785 of *LNCS*, pages 182–201. Springer, Heidelberg, March 2013.

[BDGM20a] Zvika Brakerski, Nico Döttling, Sanjam Garg, and Giulio Malavolta. Candidate io from homomorphic encryption schemes. In *EUROCRYPT 2020*, pages 79–109. Springer, 2020.

[BDGM20b] Zvika Brakerski, Nico Döttling, Sanjam Garg, and Giulio Malavolta. Factoring and pairings are not necessary for io: Circular-secure lwe suffices. *IACR Cryptology ePrint Archive*, 2020:1024, 2020.

[BDRV18] Itay Berman, Akshay Degwekar, Ron D. Rothblum, and Prashant Nalini Vasudevan. Multi-collision resistant hash functions and their applications. In Jesper Buus Nielsen and Vincent Rijmen, editors, *EUROCRYPT 2018, Part II*, volume 10821 of *LNCS*, pages 133–161. Springer, Heidelberg, April / May 2018.

[BDSG$^+$13] Nir Bitansky, Dana Dachman-Soled, Sanjam Garg, Abhishek Jain, Yael Tauman Kalai, Adriana López-Alt, and Daniel Wichs. Why “fiat-shamir for proofs” lacks a proof. In *Theory of cryptography conference*, pages 182–201. Springer, 2013.

[BDV17] Nir Bitansky, Akshay Degwekar, and Vinod Vaikuntanathan. Structure vs. hardness through the obfuscation lens. In *Annual International Cryptology Conference – CRYPTO 2017*, pages 696–723. Springer, 2017.

[BFJ$^+$20] Saikrishna Badrinarayanan, Rex Fernando, Aayush Jain, Dakshita Khurana, and Amit Sahai. Statistical ZAP arguments. In Anne Canteaut and Yuval Ishai, editors, *EUROCRYPT 2020, Part III*, volume 12107 of *LNCS*, pages 642–667. Springer, Heidelberg, May 2020.

[BFLS91] László Babai, Lance Fortnow, Leonid A. Levin, and Mario Szegedy. Checking computations in polylogarithmic time. In *23rd ACM STOC*, pages 21–31. ACM Press, May 1991.

[BFM88] M Blum, P Feldman, and S Micali. Non-interactive zero-knowledge proof systems and applications,. In *Proceedings of the 20th Annual ACM Symposium on Theory of Computing*, pages 103–112, 1988.

[BG10] Zvika Brakerski and Shafi Goldwasser. Circular and leakage resilient public-key encryption under subgroup indistinguishability. In *Annual Cryptology Conference*, pages 1–20. Springer, 2010.

[BG14] Shi Bai and Steven D. Galbraith. Lattice decoding attacks on binary LWE. In *ACISP*, volume 8544 of *Lecture Notes in Computer Science*, pages 322–337. Springer, 2014.

[BG20] Nir Bitansky and Idan Gerichter. On the cryptographic hardness of local search. In *11th Innovations in Theoretical Computer Science Conference (ITCS 2020)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2020.

[BGG90] Mihir Bellare, Oded Goldreich, and Shafi Goldwasser. Randomness in interactive proofs. In *31st FOCS*, pages 563–572. IEEE Computer Society Press, October 1990.

[BGG$^+$14] Dan Boneh, Craig Gentry, Sergey Gorbunov, Shai Halevi, Valeria Nikolaenko, Gil Segev, Vinod Vaikuntanathan, and Dhinakaran Vinayagamurthy. Fully key-homomorphic encryption, arithmetic circuit ABE and compact garbled circuits. In Phong Q. Nguyen and Elisabeth Oswald, editors, *EUROCRYPT 2014*, volume 8441 of *LNCS*, pages 533–556. Springer, Heidelberg, May 2014.

[BGI$^+$01] Boaz Barak, Oded Goldreich, Russell Impagliazzo, Steven Rudich, Amit Sahai, Salil P. Vadhan, and Ke Yang. On the (im)possibility of obfuscating programs. In Joe Kilian, editor, *CRYPTO 2001*, volume 2139 of *LNCS*, pages 1–18. Springer, Heidelberg, August 2001.

[BGI08] Eli Biham, Yaron J. Goren, and Yuval Ishai. Basing weak public-key cryptography on strong one-way functions. In *TCC*, volume 4948 of *Lecture Notes in Computer Science*, pages 55–72. Springer, 2008.

[BGI14] Elette Boyle, Shafi Goldwasser, and Ioana Ivan. Functional signatures and pseudorandom functions. In Hugo Krawczyk, editor, *PKC 2014*, volume 8383 of *LNCS*, pages 501–519. Springer, Heidelberg, March 2014.

[BGV12] Zvika Brakerski, Craig Gentry, and Vinod Vaikuntanathan. (Leveled) fully homomorphic encryption without bootstrapping. In Shafi Goldwasser, editor, *ITCS 2012*, pages 309–325. ACM, January 2012.

[BHHI10] Boaz Barak, Iftach Haitner, Dennis Hofheinz, and Yuval Ishai. Bounded key-dependent message security. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT)*, pages 423–444. Springer, 2010.

[BHHO08] Dan Boneh, Shai Halevi, Mike Hamburg, and Rafail Ostrovsky. Circular-secure encryption from decision diffie-hellman. In *Annual International Cryptology Conference*, pages 108–125. Springer, 2008.

[BHK17] Zvika Brakerski, Justin Holmgren, and Yael Tauman Kalai. Non-interactive delegation and batch NP verification from standard computational assumptions. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2017, Montreal, QC, Canada, June 19-23, 2017*, pages 474–482, 2017.

[BHY09] Mihir Bellare, Dennis Hofheinz, and Scott Yilek. Possibility and impossibility results for encryption and commitment secure under selective opening. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 1–35. Springer, 2009.

[BKM17] Dan Boneh, Sam Kim, and Hart William Montgomery. Private puncturable PRFs from standard lattice assumptions. In Jean-Sébastien Coron and Jesper Buus Nielsen, editors, *EUROCRYPT 2017, Part I*, volume 10210 of *LNCS*, pages 415–445. Springer, Heidelberg, April / May 2017.

[BKM20] Zvika Brakerski, Venkata Koppula, and Tamer Mour. NIZK from LPN and trapdoor hash via correlation intractability for approximable relations. In Daniele Micciancio and Thomas Ristenpart, editors, *CRYPTO 2020, Part III*, volume 12172 of *LNCS*, pages 738–767. Springer, Heidelberg, August 2020.

[BKP18] Nir Bitansky, Yael Tauman Kalai, and Omer Paneth. Multi-collision resistance: a paradigm for keyless hash functions. In Ilias Diakonikolas, David Kempe, and Monika Henzinger, editors, *50th ACM STOC*, pages 671–684. ACM Press, June 2018.

[BKW03] Avrim Blum, Adam Kalai, and Hal Wasserman. Noise-tolerant learning, the parity problem, and the statistical query model. *Journal of the ACM (JACM)*, 50(4):506–519, 2003.

[BL18] Nir Bitansky and Huijia Lin. One-message zero knowledge and non-malleable commitments. 2018.

[BLMR13] Dan Boneh, Kevin Lewi, Hart Montgomery, and Ananth Raghunathan. Key homomorphic prfs and their applications. In *Advances in Cryptology–CRYPTO 2013*, pages 410–428. Springer, 2013.

[BLP$^+$13] Zvika Brakerski, Adeline Langlois, Chris Peikert, Oded Regev, and Damien Stehlé. Classical hardness of learning with errors. In *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*, pages 575–584. ACM, 2013.

[BLS16] Shi Bai, Thijs Laarhoven, and Damien Stehlé. Tuple lattice sieving. *LMS Journal of Computation and Mathematics*, 19(A):146–162, 2016.

[BLSV18] Zvika Brakerski, Alex Lombardi, Gil Segev, and Vinod Vaikuntanathan. Anonymous IBE, leakage resilience and circular security from new assumptions. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 535–564. Springer, 2018.

[Blu86] Manuel Blum. How to prove a theorem so no one else can claim it. In *Proceedings of the International Congress of Mathematicians*, volume 1, page 2. Citeseer, 1986.

[BLV03] Boaz Barak, Yehuda Lindell, and Salil P. Vadhan. Lower bounds for non-black-box zero knowledge. In *44th FOCS*, pages 384–393. IEEE Computer Society Press, October 2003.

[BLVW18] Zvika Brakerski, Vadim Lyubashevsky, Vinod Vaikuntanathan, and Daniel Wichs. Cryptographic hashing and worst-case hardness for lpn via code smoothing. 2018.

[BLW17] Dan Boneh, Kevin Lewi, and David J. Wu. Constraining pseudorandom functions privately. In Serge Fehr, editor, *PKC 2017, Part II*, volume 10175 of *LNCS*, pages 494–524. Springer, Heidelberg, March 2017.

[BM82] Manuel Blum and Silvio Micali. How to generate cryptographically strong sequences of pseudo random bits. In *23rd FOCS*, pages 112–117. IEEE Computer Society Press, November 1982.

[BMR90] Donald Beaver, Silvio Micali, and Phillip Rogaway. The round complexity of secure protocols. In *Proceedings of the twenty-second annual ACM symposium on Theory of computing*, pages 503–513. ACM, 1990.

[BP15] Nir Bitansky and Omer Paneth. ZAPs and Non-Interactive Witness Indistinguishability from Indistinguishability Obfuscation. In *Theory of Cryptography - TCC 2015*, 2015.

[BPR15] Nir Bitansky, Omer Paneth, and Alon Rosen. On the cryptographic hardness of finding a nash equilibrium. In *FOCS 2015*. IEEE, 2015.

[BPW16] Nir Bitansky, Omer Paneth, and Daniel Wichs. Perfect structure on the edge of chaos - trapdoor permutations from indistinguishability obfuscation. In Eyal Kushilevitz and Tal Malkin, editors, *TCC 2016-A, Part I*, volume 9562 of *LNCS*, pages 474–502. Springer, Heidelberg, January 2016.

[BR93] Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In Dorothy E. Denning, Raymond Pyle, Ravi Ganesan, Ravi S. Sandhu, and Victoria Ashby, editors, *ACM CCS 93*, pages 62–73. ACM Press, November 1993.

[Bra12] Zvika Brakerski. Fully homomorphic encryption without modulus switching from classical GapSVP. In *Advances in Cryptology–CRYPTO 2012*, pages 868–886. Springer, 2012.

[BRT12] Mihir Bellare, Thomas Ristenpart, and Stefano Tessaro. Multi-instance security and its application to password-based cryptography. In *CRYPTO*, volume 7417 of *Lecture Notes in Computer Science*, pages 312–329. Springer, 2012.

[BV11] Zvika Brakerski and Vinod Vaikuntanathan. Efficient fully homomorphic encryption from (standard) LWE. In Rafail Ostrovsky, editor, *52nd FOCS*, pages 97–106. IEEE Computer Society Press, October 2011.

[BV14] Zvika Brakerski and Vinod Vaikuntanathan. Lattice-based FHE as secure as PKE. In Moni Naor, editor, *ITCS 2014*, pages 1–12. ACM, January 2014.

[BV15] Zvika Brakerski and Vinod Vaikuntanathan. Constrained key-homomorphic PRFs from standard lattice assumptions - or: How to secretly embed a circuit in your PRF. In Yevgeniy Dodis and Jesper Buus Nielsen, editors, *TCC 2015, Part II*, volume 9015 of *LNCS*, pages 1–30. Springer, Heidelberg, March 2015.

[BW13] Dan Boneh and Brent Waters. Constrained pseudorandom functions and their applications. In Kazue Sako and Palash Sarkar, editors, *ASIACRYPT 2013, Part II*, volume 8270 of *LNCS*, pages 280–300. Springer, Heidelberg, December 2013.

[Can97] Ran Canetti. Towards realizing random oracles: Hash functions that hide all partial information. *IACR Cryptology ePrint Archive*, 1997:7, 1997.

[CC17] Ran Canetti and Yilei Chen. Constraint-hiding constrained prfs for nc1 from lwe. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 446–476. Springer, 2017.

[CCH+18] Ran Canetti, Yilei Chen, Justin Holmgren, Alex Lombardi, Guy N Rothblum, and Ron D Rothblum. Fiat-Shamir from simpler assumptions. *IACR Cryptology ePrint Archive*, 2018:1004, 2018. `https://eprint.iacr.org/2018/1004`. Part I of [CCH+19].

[CCH+19] Ran Canetti, Yilei Chen, Justin Holmgren, Alex Lombardi, Guy N. Rothblum, Ron D. Rothblum, and Daniel Wichs. Fiat-Shamir: from practice to theory. In Moses Charikar and Edith Cohen, editors, *51st ACM STOC*, pages 1082–1090. ACM Press, June 2019.

[CCR16] Ran Canetti, Yilei Chen, and Leonid Reyzin. On the correlation intractability of obfuscated pseudorandom functions. In Eyal Kushilevitz and Tal Malkin, editors, *TCC 2016-A, Part I*, volume 9562 of *LNCS*, pages 389–415. Springer, Heidelberg, January 2016.

[CCRR18] Ran Canetti, Yilei Chen, Leonid Reyzin, and Ron D. Rothblum. Fiat-Shamir and correlation intractability from strong KDM-secure encryption. In Jesper Buus Nielsen and Vincent Rijmen, editors, *EUROCRYPT 2018, Part I*, volume 10820 of *LNCS*, pages 91–122. Springer, Heidelberg, April / May 2018.

[CDT09] Xi Chen, Xiaotie Deng, and Shang-Hua Teng. Settling the complexity of computing two-player nash equilibria. *Journal of the ACM (JACM)*, 56(3):1–57, 2009.

[CEP83] E Rodney Canfield, Paul Erdös, and Carl Pomerance. On a problem of oppenheim concerning “factorisatio numerorum”. *Journal of Number Theory*, 17(1):1–28, 1983.

[CGGM00] Ran Canetti, Oded Goldreich, Shafi Goldwasser, and Silvio Micali. Resettable zero-knowledge. In *STOC 2000*, pages 235–244, 2000.

[CGH98] Ran Canetti, Oded Goldreich, and Shai Halevi. The random oracle methodology, revisited (preliminary version). In *30th ACM STOC*, pages 209–218. ACM Press, May 1998.

[CHK03] Ran Canetti, Shai Halevi, and Jonathan Katz. A forward-secure public-key encryption scheme. *IACR Cryptology ePrint Archive*, 2003:83, 2003.

[CHK$^+$19a] Arka Rai Choudhuri, Pavel Hubácek, Chethan Kamath, Krzysztof Pietrzak, Alon Rosen, and Guy N. Rothblum. Finding a nash equilibrium is no easier than breaking Fiat-Shamir. In Moses Charikar and Edith Cohen, editors, *51st ACM STOC*, pages 1103–1114. ACM Press, June 2019.

[CHK$^+$19b] Arka Rai Choudhuri, Pavel Hubácek, Chethan Kamath, Krzysztof Pietrzak, Alon Rosen, and Guy N Rothblum. Ppad-hardness via iterated squaring modulo a composite. Cryptology ePrint Archive, Report 2019/667, 2019., 2019.

[CJJ21a] Arka Rai Choudhuri, Abhishek Jain, and Zhengzhong Jin. Non-interactive batch arguments for NP from standard assumptions. In Tal Malkin and Chris Peikert, editors, *CRYPTO 2021, Part IV*, volume 12828 of *LNCS*, pages 394–423, Virtual Event, August 2021. Springer, Heidelberg.

[CJJ21b] Arka Rai Choudhuri, Abhishek Jain, and Zhengzhong Jin. SNARGs for P from LWE. 2021. Proceedings of FOCS 2021, to appear.

[CK16] Aloni Cohen and Saleet Klein. The GGM function family is a weakly one-way family of functions. In Martin Hirt and Adam D. Smith, editors, *TCC 2016-B, Part I*, volume 9985 of *LNCS*, pages 84–107. Springer, Heidelberg, October / November 2016.

[CK18] Henry Corrigan-Gibbs and Dmitry Kogan. The discrete-logarithm problem with preprocessing. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques – EUROCRYPT 2018*, 2018.

[CLMQ21] Yilei Chen, Alex Lombardi, Fermi Ma, and Willy Quach. Does fiat-shamir require a cryptographic hash function? In Tal Malkin and Chris Peikert, editors, *CRYPTO 2021, Part IV*, volume 12828 of *LNCS*, pages 334–363, Virtual Event, August 2021. Springer, Heidelberg.

[CLTV15] Ran Canetti, Huijia Lin, Stefano Tessaro, and Vinod Vaikuntanathan. Obfuscation of probabilistic circuits and applications. In *Theory of Cryptography Conference*, pages 468–497. Springer, 2015.

[CLW18] Ran Canetti, Alex Lombardi, and Daniel Wichs. Fiat-Shamir: From practice to theory, part II (NIZK and correlation intractability from circular-secure FHE). *Cryptology ePrint Archive*, 2018. https://eprint.iacr.org/2018/1248. Part II of [CCH+19].

[CM19] Jing Chen and Silvio Micali. Algorand: A secure and efficient distributed ledger. *Theoretical Computer Science*, 777:155–183, 2019.

[CMR98] Ran Canetti, Daniele Micciancio, and Omer Reingold. Perfectly one-way probabilistic hash functions (preliminary version). In *30th ACM STOC*, pages 131–140. ACM Press, May 1998.

[CN11] Yuanmi Chen and Phong Q Nguyen. BKZ 2.0: Better lattice security estimates. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 1–20. Springer, 2011.

[CZ81] David G Cantor and Hans Zassenhaus. A new algorithm for factoring polynomials over finite fields. *Mathematics of Computation*, pages 587–592, 1981.

[Dam88] Ivan Damgård. Collision free hash functions and public key signature schemes. In David Chaum and Wyn L. Price, editors, *EUROCRYPT'87*, volume 304 of *LNCS*, pages 203–216. Springer, Heidelberg, April 1988.

[dFMPS19] Luca de Feo, Simon Masson, Christophe Petit, and Antonio Sanso. Verifiable delay functions from supersingular isogenies and pairings. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 248–277. Springer, 2019.

[DGP06] Constantinos Daskalakis, Paul W. Goldberg, and Christos H. Papadimitriou. The complexity of computing a Nash equilibrium. In Jon M. Kleinberg, editor, *38th ACM STOC*, pages 71–78. ACM Press, May 2006.

[DH76] Whitfield Diffie and Martin E Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, 22(6), 1976.

[Dix81] John D Dixon. Asymptotically fast factorization of integers. *Mathematics of computation*, 36(153):255–260, 1981.

[DJMW12] Yevgeniy Dodis, Abhishek Jain, Tal Moran, and Daniel Wichs. Counterexamples to hardness amplification beyond negligible. In *Theory of Cryptography Conference*, pages 476–493. Springer, 2012.

[DN00] Cynthia Dwork and Moni Naor. Zaps and their applications. In *FOCS 2000*, pages 283–293. IEEE, 2000.

[DN01] Ivan Damgård and Jesper Buus Nielsen. Perfect hiding and perfect binding universally composable commitment schemes with constant expansion factor. *IACR Cryptology ePrint Archive*, 2001:91, 2001.

[DNRS99] Cynthia Dwork, Moni Naor, Omer Reingold, and Larry J. Stockmeyer. Magic functions. In *40th FOCS*, pages 523–534. IEEE Computer Society Press, October 1999.

[DP11] Constantinos Daskalakis and Christos Papadimitriou. Continuous local search. In *SODA 2011*, pages 790–804. SIAM, 2011.

[DS11] Yevgeniy Dodis and John P. Steinberger. Domain extension for MACs beyond the birthday barrier. In Kenneth G. Paterson, editor, *EUROCRYPT 2011*, volume 6632 of *LNCS*, pages 323–342. Springer, Heidelberg, May 2011.

[DVW20] Yevgeniy Dodis, Vinod Vaikuntanathan, and Daniel Wichs. Extracting randomness from extractor-dependent sources. In Anne Canteaut and Yuval Ishai, editors, *EUROCRYPT 2020, Part I*, volume 12105 of *LNCS*, pages 313–342. Springer, Heidelberg, May 2020.

[DW20] Dean Doron and Mary Wootters. High-probability list-recovery, and applications to heavy hitters. *ECCC*, 2020. https://eccc.weizmann.ac.il/report/2020/162/.

[EFKP19] Naomi Ephraim, Cody Freitag, Ilan Komargodski, and Rafael Pass. Continuous verifiable delay functions. In *EUROCRYPT 2020*, 2019.

[Fab19] After 20 years, someone finally solved this mit puzzle, 2019.

[FGJ18] Nils Fleischhacker, Vipul Goyal, and Abhishek Jain. On the existence of three round zero-knowledge proofs. In Jesper Buus Nielsen and Vincent Rijmen, editors, *EUROCRYPT 2018, Part III*, volume 10822 of *LNCS*, pages 3–33. Springer, Heidelberg, April / May 2018.

[FGL$^+$91] Uriel Feige, Shafi Goldwasser, László Lovász, Shmuel Safra, and Mario Szegedy. Approximating clique is almost NP-complete (preliminary version). In *32nd FOCS*, pages 2–12. IEEE Computer Society Press, October 1991.

[FLS90] Uriel Feige, Dror Lapidot, and Adi Shamir. Multiple non-interactive zero knowledge proofs based on a single random string (extended abstract). In *31st FOCS*, pages 308–317. IEEE Computer Society Press, October 1990.

[For66] G David Forney. Concatenated codes. 1966.

[FS87] Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In Andrew M. Odlyzko, editor, *CRYPTO'86*, volume 263 of *LNCS*, pages 186–194. Springer, Heidelberg, August 1987.

[Gen09] Craig Gentry. Fully homomorphic encryption using ideal lattices. In Michael Mitzenmacher, editor, *41st ACM STOC*, pages 169–178. ACM Press, May / June 2009.

[GGH^{+}13] Sanjam Garg, Craig Gentry, Shai Halevi, Mariana Raykova, Amit Sahai, and Brent Waters. Candidate Indistinguishability Obfuscation and Functional Encryption for all Circuits. In *Foundations of Computer Science (FOCS), 2013 IEEE 54th Annual Symposium on*, pages 40–49, 2013.

[GGM84] Oded Goldreich, Shafi Goldwasser, and Silvio Micali. How to construct random functions (extended abstract). In *25th FOCS*, pages 464–479. IEEE Computer Society Press, October 1984.

[GI01] Venkatesan Guruswami and Piotr Indyk. Expander-based constructions of efficiently decodable codes. In *42nd FOCS*, pages 658–667. IEEE Computer Society Press, October 2001.

[GI02] Venkatesan Guruswami and Piotr Indyk. Near-optimal linear-time codes for unique decoding and new list-decodable codes over smaller alphabets. In *34th ACM STOC*, pages 812–821. ACM Press, May 2002.

[GI03] Venkatesan Guruswami and Piotr Indyk. Linear time encodable and list decodable codes. In *35th ACM STOC*, pages 126–135. ACM Press, June 2003.

[GI04] Venkatesan Guruswami and Piotr Indyk. Linear-time list decoding in error-free settings: (extended abstract). In Josep Díaz, Juhani Karhumäki, Arto Lepistö, and Donald Sannella, editors, *ICALP 2004*, volume 3142 of *LNCS*, pages 695–707. Springer, Heidelberg, July 2004.

[GJJM20] Vipul Goyal, Abhishek Jain, Zhengzhong Jin, and Giulio Malavolta. Statistical zaps and new oblivious transfer protocols. In Anne Canteaut and Yuval Ishai, editors, *EUROCRYPT 2020, Part III*, volume 12107 of *LNCS*, pages 668–699. Springer, Heidelberg, May 2020.

[GK96] Oded Goldreich and Hugo Krawczyk. On the composition of zero-knowledge proof systems. *SIAM Journal on Computing*, 25(1):169–192, 1996.

[GK03] Shafi Goldwasser and Yael Tauman Kalai. On the (in)security of the Fiat-Shamir paradigm. In *44th FOCS*, pages 102–115. IEEE Computer Society Press, October 2003.

[GK16] Shafi Goldwasser and Yael Tauman Kalai. Cryptographic assumptions: A position paper. In *Theory of Cryptography Conference*, pages 505–522. Springer, 2016.

[GKP$^+$13] Shafi Goldwasser, Yael Kalai, Raluca Ada Popa, Vinod Vaikuntanathan, and Nickolai Zeldovich. Reusable garbled circuits and succinct functional encryption. In *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*, pages 555–564. ACM, 2013.

[GKR08] Shafi Goldwasser, Yael Tauman Kalai, and Guy N. Rothblum. Delegating computation: interactive proofs for muggles. In Richard E. Ladner and Cynthia Dwork, editors, *40th ACM STOC*, pages 113–122. ACM Press, May 2008.

[GKW17a] Rishab Goyal, Venkata Koppula, and Brent Waters. Lockable obfuscation. In *Foundations of Computer Science (FOCS), 2017 IEEE 58th Annual Symposium on*, pages 612–621. IEEE, 2017.

[GKW17b] Rishab Goyal, Venkata Koppula, and Brent Waters. Separating semantic and circular security for symmetric-key bit encryption from the learning with errors assumption. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 528–557. Springer, 2017.

[GKW18] Rishab Goyal, Venkata Koppula, and Brent Waters. Collusion resistant traitor tracing from learning with errors. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, pages 660–670. ACM, 2018.

[GM84] Shafi Goldwasser and Silvio Micali. Probabilistic encryption. *Journal of computer and system sciences*, 28(2):270–299, 1984.

[GMR85] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proof-systems (extended abstract). In *17th ACM STOC*, pages 291–304. ACM Press, May 1985.

[GMW86] Oded Goldreich, Silvio Micali, and Avi Wigderson. Proofs that yield nothing but their validity and a methodology of cryptographic protocol design (extended abstract). In *27th FOCS*, pages 174–187. IEEE Computer Society Press, October 1986.

[GMW87] Oded Goldreich, Silvio Micali, and Avi Wigderson. How to play any mental game or A completeness theorem for protocols with honest majority. In Alfred Aho, editor, *19th ACM STOC*, pages 218–229. ACM Press, May 1987.

[GN08] Nicolas Gama and Phong Q. Nguyen. Predicting lattice reduction. In *EUROCRYPT*, volume 4965 of *Lecture Notes in Computer Science*, pages 31–51. Springer, 2008.

[GO94] Oded Goldreich and Yair Oren. Definitions and properties of zero-knowledge proof systems. *Journal of Cryptology*, 7(1):1–32, December 1994.

[Gol99] Oded Goldreich. *Modern cryptography, probabilistic proofs and pseudorandomness*, volume 17 of *Algorithms and Combinatorics*. Springer-Verlag, 1999.

[Gol04] Oded Goldreich. *The Foundations of Cryptography - Volume 2, Basic Applications*. Cambridge University Press, 2004.

[Gol07] Oded Goldreich. *Foundations of cryptography: volume 1, basic tools*. Cambridge university press, 2007.

[Gol11] Oded Goldreich. A sample of samplers: A computational perspective on sampling. In *Studies in Complexity and Cryptography. Miscellanea on the Interplay between Randomness and Computation*, pages 302–332. Springer, 2011.

[Gol17] Oded Goldreich. On the doubly-efficient interactive proof systems of GKR. *Electronic Colloquium on Computational Complexity (ECCC)*, 24:101, 2017.

[GOS06] Jens Groth, Rafail Ostrovsky, and Amit Sahai. Non-interactive zaps and new techniques for nizk. In *Annual International Cryptology Conference*, pages 97–111. Springer, 2006.

[GP21] Romain Gay and Rafael Pass. Indistinguishability obfuscation from circular security. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, pages 736–749, 2021.

[GPS16] Sanjam Garg, Omkant Pandey, and Akshayaram Srinivasan. Revisiting the cryptographic hardness of finding a nash equilibrium. In *CRYPTO*, pages 579–604. Springer, 2016.

[GPV08] Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In *Proceedings of the fortieth annual ACM symposium on Theory of computing*, pages 197–206. ACM, 2008.

[GR07] Shafi Goldwasser and Guy N Rothblum. On best-possible obfuscation. In *Theory of Cryptography Conference*, pages 194–213. Springer, 2007.

[GR08] Venkatesan Guruswami and Atri Rudra. Soft decoding, dual bch codes, and better list-decodable e-biased codes. In *2008 23rd Annual IEEE Conference on Computational Complexity*, pages 163–174. IEEE, 2008.

[GR13] Oded Goldreich and Ron D. Rothblum. Enhancements of trapdoor permutations. *Journal of Cryptology*, 26(3):484–512, July 2013.

[Gra08] Andrew Granville. Smooth numbers: computational number theory and beyond. *Algorithmic number theory: lattices, number fields, curves and cryptography*, 44:267–323, 2008.

[GS86] Shafi Goldwasser and Michael Sipser. Private coins versus public coins in interactive proof systems. In *18th ACM STOC*, pages 59–68. ACM Press, May 1986.

[GS98] Venkatesan Guruswami and Madhu Sudan. Improved decoding of Reed-Solomon and algebraic-geometric codes. In *39th FOCS*, pages 28–39. IEEE Computer Society Press, November 1998.

[GSW13] Craig Gentry, Amit Sahai, and Brent Waters. Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based. In Ran Canetti and Juan A. Garay, editors, *CRYPTO 2013, Part I*, volume 8042 of *LNCS*, pages 75–92. Springer, Heidelberg, August 2013.

[GUV09] Venkatesan Guruswami, Christopher Umans, and Salil Vadhan. Unbalanced expanders and randomness extractors from parvaresh–vardy codes. *Journal of the ACM (JACM)*, 56(4):1–34, 2009.

[GVW13] Sergey Gorbunov, Vinod Vaikuntanathan, and Hoeteck Wee. Attribute-based encryption for circuits. In *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*, pages 545–554. ACM, 2013.

[GVW15] Sergey Gorbunov, Vinod Vaikuntanathan, and Hoeteck Wee. Predicate encryption for circuits from LWE. In Rosario Gennaro and Matthew J. B. Robshaw, editors, *CRYPTO 2015, Part II*, volume 9216 of *LNCS*, pages 503–523. Springer, Heidelberg, August 2015.

[GW11] Craig Gentry and Daniel Wichs. Separating succinct non-interactive arguments from all falsifiable assumptions. In Lance Fortnow and Salil P. Vadhan, editors, *43rd ACM STOC*, pages 99–108. ACM Press, June 2011.

[HILL99] Johan Håstad, Russell Impagliazzo, Leonid A Levin, and Michael Luby. A pseudorandom generator from any one-way function. *SIAM Journal on Computing*, 28(4):1364–1396, 1999.

[HIOS15] Iftach Haitner, Yuval Ishai, Eran Omri, and Ronen Shaltiel. Parallel hashing via list recoverability. In Rosario Gennaro and Matthew J. B. Robshaw, editors, *CRYPTO 2015, Part II*, volume 9216 of *LNCS*, pages 173–190. Springer, Heidelberg, August 2015.

[HJKS22] James Hulett, Ruta Jawale, Dakshita Khurana, and Akshayaram Srinivasan. Snargs for p from sub-exponential ddh and qr. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 520–549. Springer, 2022.

[HL18] Justin Holmgren and Alex Lombardi. Cryptographic hashing from strong one-way functions (or: One-way product functions and their applications). In Mikkel Thorup, editor, *59th FOCS*, pages 850–858. IEEE Computer Society Press, October 2018.

[HLR21] Justin Holmgren, Alex Lombardi, and Ron D. Rothblum. Fiat-Shamir via list-recoverable codes (or: Parallel repetition of GMW is not zero-knowledge). Cryptology ePrint Archive, Report 2021/286, 2021. https://eprint.iacr.org/2021/286. Proceedings of STOC 2021.

[HMR08] Shai Halevi, Steven Myers, and Charles Rackoff. On seed-incompressible functions. In *Theory of Cryptography Conference*, pages 19–36. Springer, 2008.

[HU19] Dennis Hofheinz and Bogdan Ursu. Dual-mode NIZKs from obfuscation. In Steven D. Galbraith and Shiho Moriai, editors, *ASIACRYPT 2019, Part I*, volume 11921 of *LNCS*, pages 311–341. Springer, Heidelberg, December 2019.

[HW15a] Brett Hemenway and Mary Wootters. Linear-time list recovery of high-rate expander codes. In Magnús M. Halldórsson, Kazuo Iwama, Naoki Kobayashi, and Bettina Speckmann, editors, *ICALP 2015, Part I*, volume 9134 of *LNCS*, pages 701–712. Springer, Heidelberg, July 2015.

[HW15b] Pavel Hubacek and Daniel Wichs. On the communication complexity of secure function evaluation with long output. In *Proceedings of the 2015 Conference on Innovations in Theoretical Computer Science*, pages 163–172. ACM, 2015.

[HY17] Pavel Hubáček and Eylon Yogev. Hardness of continuous local search: Query complexity and cryptographic lower bounds. In *SODA 2017*, pages 1352–1371. SIAM, 2017.

[IK02] Yuval Ishai and Eyal Kushilevitz. Perfect constant-round secure computation via perfect randomizing polynomials. In *International Colloquium on Automata, Languages, and Programming*, pages 244–256. Springer, 2002.

[IKOS07] Yuval Ishai, Eyal Kushilevitz, Rafail Ostrovsky, and Amit Sahai. Zero-knowledge from secure multiparty computation. In David S. Johnson and Uriel Feige, editors, *39th ACM STOC*, pages 21–30. ACM Press, June 2007.

[Ish20] Yuval Ishai. Zero-knowledge proofs from information-theoretic proof systems. 2020. https://zkproof.org/2020/08/12/information-theoretic-proof-systems/.

[IW97] Russell Impagliazzo and Avi Wigderson. P = BPP if E requires exponential circuits: Derandomizing the XOR lemma. In *29th ACM STOC*, pages 220–229. ACM Press, May 1997.

[JJ19] Abhishek Jain and Zhengzhong Jin. Statistical zap arguments from quasi-polynomial LWE. Cryptology ePrint Archive, Report 2019/839, 2019. https://eprint.iacr.org/2019/839.

[JJ21] Abhishek Jain and Zhengzhong Jin. Non-interactive zero knowledge from sub-exponential DDH. In Anne Canteaut and François-Xavier Standaert, editors, *EUROCRYPT 2021, Part I*, volume 12696 of *LNCS*, pages 3–32. Springer, Heidelberg, October 2021.

[JKKZ21] Ruta Jawale, Yael Tauman Kalai, Dakshita Khurana, and Rachel Zhang. SNARGs for bounded depth computations and ppad hardness from sub-exponential LWE. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing (STOC)*, pages 708–721, 2021.

[JLMS19] Aayush Jain, Huijia Lin, Christian Matt, and Amit Sahai. How to leverage hardness of constant-degree expanding polynomials over r to build io. In *Proceedings of EUROCRYPT 2019*, 2019.

[JLS19] Aayush Jain, Huijia Lin, and Amit Sahai. Simplifying constructions and assumptions for $i\mathcal{O}$. Cryptology ePrint Archive, Report 2019/1252, 2019.

[JLS21] Aayush Jain, Huijia Lin, and Amit Sahai. Indistinguishability obfuscation from well-founded assumptions. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, pages 60–73, 2021.

[JOP14] Antoine Joux, Andrew Odlyzko, and Cécile Pierrot. The past, evolving present, and future of the discrete logarithm. In *Open Problems in Mathematics and Computational Science*, pages 5–36. Springer, 2014.

[Kan83] Ravi Kannan. Improved algorithms for integer programming and related lattice problems. In *STOC 1983*, pages 193–206, 1983.

[Kan87] Ravi Kannan. Minkowski's convex body theorem and integer programming. *Mathematics of operations research*, 12(3):415–440, 1987.

[KF16] Paul Kirchner and Pierre-Alain Fouque. Time-memory trade-off for lattice enumeration in a ball. *IACR Cryptology ePrint Archive*, 2016:222, 2016.

[Kil92] Joe Kilian. A note on efficient zero-knowledge proofs and arguments (extended abstract). In *24th ACM STOC*, pages 723–732. ACM Press, May 1992.

[KN08] Gillat Kol and Moni Naor. Cryptography and game theory: Designing protocols for exchanging information. In *Theory of Cryptography Conference*, pages 320–339. Springer, 2008.

[KNY17] Ilan Komargodski, Moni Naor, and Eylon Yogev. White-box vs. black-box complexity of search problems: Ramsey and graph property testing. In Chris Umans, editor, *58th FOCS*, pages 622–632. IEEE Computer Society Press, October 2017.

[KNY18] Ilan Komargodski, Moni Naor, and Eylon Yogev. Collision resistant hashing for paranoids: Dealing with multiple collisions. In Jesper Buus Nielsen and Vincent Rijmen, editors, *EUROCRYPT 2018, Part II*, volume 10821 of *LNCS*, pages 162–194. Springer, Heidelberg, April / May 2018.

[KPTZ13] Aggelos Kiayias, Stavros Papadopoulos, Nikos Triandopoulos, and Thomas Zacharias. Delegatable pseudorandom functions and applications. In Ahmad-Reza Sadeghi, Virgil D. Gligor, and Moti Yung, editors, *ACM CCS 2013*, pages 669–684. ACM Press, November 2013.

[KPY18] Yael Kalai, Omer Paneth, and Lisa Yang. On publicly verifiable delegation from standard assumptions. *IACR Cryptology ePrint Archive*, 2018:776, 2018.

[KPY19] Yael Kalai, Omer Paneth, and Lisa Yang. How to delegate computations publicly. In *Proceedings of the fifty-first annual ACM Symposium on Theory of Computing*, volume 2019, 2019.

[KPY20] Yael Kalai, Omer Paneth, and Lisa Yang. Ppad-hardness and delegation with unambiguous and updatable proofs. In *CRYPTO 2020*, 2020.

[KRR14] Yael Tauman Kalai, Ran Raz, and Ron D. Rothblum. How to delegate computations: the power of no-signaling proofs. In *Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014*, pages 485–494, 2014.

[KRR17] Yael Tauman Kalai, Guy N. Rothblum, and Ron D. Rothblum. From obfuscation to the security of Fiat-Shamir for proofs. In Jonathan Katz and Hovav Shacham, editors, *CRYPTO 2017, Part II*, volume 10402 of *LNCS*, pages 224–251. Springer, Heidelberg, August 2017.

[KV16] Seungki Kim and Akshay Venkatesh. The behavior of random reduced bases. *International Mathematics Research Notices*, 2016.

[KVZ21] Yael Tauman Kalai, Vinod Vaikuntanathan, and Rachel Yun Zhang. Somewhere statistical soundness, post-quantum security, and snargs. In *Theory of Cryptography Conference*, pages 330–368. Springer, 2021.

[Lev73] Leonid Anatolevich Levin. Universal sequential search problems. *Problemy Peredachi Informatsii*, 9(3):115–116, 1973.

[LFKN90] Carsten Lund, Lance Fortnow, Howard J. Karloff, and Noam Nisan. Algebraic methods for interactive proof systems. In *31st FOCS*, pages 2–10. IEEE Computer Society Press, October 1990.

[LLL82] Arjen Klaas Lenstra, Hendrik Willem Lenstra, and László Lovász. Factoring polynomials with rational coefficients. *Mathematische Annalen*, 261(4):515–534, 1982.

[LLMP90] Arjen K Lenstra, Hendrik W Lenstra Jr, Mark S Manasse, and John M Pollard. The number field sieve. In *STOC 1990*, pages 564–572, 1990.

[LNPT19] Benoît Libert, Khoa Nguyen, Alain Passelègue, and Radu Titiu. Simulation-sound arguments for LWE and applications to KDM-CCA2 security. Cryptology ePrint Archive, Report 2019/908, 2019. https://eprint.iacr.org/2019/908.

[LNPY20] Benoît Libert, Khoa Nguyen, Thomas Peters, and Moti Yung. One-shot fiat-shamir-based nizk arguments of composite residuosity in the standard model. Cryptology ePrint Archive, Report 2020/1334, 2020. https://eprint.iacr.org/2020/1334.

[LP09] Yehuda Lindell and Benny Pinkas. A proof of security of yao's protocol for two-party computation. *Journal of Cryptology*, 22(2):161–188, 2009.

[LV20a] Alex Lombardi and Vinod Vaikuntanathan. Fiat-shamir for repeated squaring with applications to PPAD-hardness and VDFs. In Daniele Micciancio and Thomas Ristenpart, editors, *CRYPTO 2020, Part III*, volume 12172 of *LNCS*, pages 632–651. Springer, Heidelberg, August 2020.

[LV20b] Alex Lombardi and Vinod Vaikuntanathan. Multi-input correlation intractable hash functions via shift-hiding, 2020. To appear in ITCS 2022. https://eprint.iacr.org/2020/1378.

[LVW19] Alex Lombardi, Vinod Vaikuntanathan, and Daniel Wichs. 2-message publicly verifiable WI from (subexponential) LWE. Cryptology ePrint Archive, Report 2019/808, 2019. https://eprint.iacr.org/2019/808.

[Mah18] Urmila Mahadev. Classical verification of quantum computations. In Mikkel Thorup, editor, *59th FOCS*, pages 259–267. IEEE Computer Society Press, October 2018.

[Mer79] Ralph Charles Merkle. *Secrecy, authentication, and public key systems.* Stanford university, 1979.

[Mer88] Ralph C. Merkle. A digital signature based on a conventional encryption function. In Carl Pomerance, editor, *CRYPTO'87*, volume 293 of *LNCS*, pages 369–378. Springer, Heidelberg, August 1988.

[Mic94] Silvio Micali. CS proofs (extended abstracts). In *35th FOCS*, pages 436–453. IEEE Computer Society Press, November 1994.

[Mou21] Tamer Mour. Correlation intractability vs. one-wayness. *Cryptology ePrint Archive*, 2021.

[MP13] Daniele Micciancio and Chris Peikert. Hardness of SIS and LWE with small parameters. In *Advances in Cryptology–CRYPTO 2013*, pages 21–39. Springer, 2013.

[MR09] Daniele Micciancio and Oded Regev. Lattice-based cryptography. In *Post-quantum cryptography*, pages 147–191. Springer, 2009.

[MT07] Ueli M. Maurer and Stefano Tessaro. Domain extension of public random functions: Beyond the birthday barrier. In Alfred Menezes, editor, *CRYPTO 2007*, volume 4622 of *LNCS*, pages 187–204. Springer, Heidelberg, August 2007.

[MW16] Daniele Micciancio and Michael Walter. Practical, predictable lattice basis reduction. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 820–849. Springer, 2016.

[Nak08] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. *Decentralized Business Review*, page 21260, 2008.

[Nao91] Moni Naor. Bit commitment using pseudorandomness. *Journal of cryptology*, 4(2):151–158, 1991.

[Nao03] Moni Naor. On cryptographic assumptions and challenges (invited talk). In Dan Boneh, editor, *CRYPTO 2003*, volume 2729 of *LNCS*, pages 96–109. Springer, Heidelberg, August 2003.

[NS06] Phong Q. Nguyen and Damien Stehlé. LLL on the average. In *ANTS*, volume 4076 of *Lecture Notes in Computer Science*, pages 238–256. Springer, 2006.

[NW88] Noam Nisan and Avi Wigderson. Hardness vs. randomness (extended abstract). In *29th FOCS*, pages 2–11. IEEE Computer Society Press, October 1988.

[NY89] Moni Naor and Moti Yung. Universal one-way hash functions and their cryptographic applications. In *21st ACM STOC*, pages 33–43. ACM Press, May 1989.

[Oka93] Tatsuaki Okamoto. Provably secure and practical identification schemes and corresponding signature schemes. In Ernest F. Brickell, editor, *CRYPTO'92*, volume 740 of *LNCS*, pages 31–53. Springer, Heidelberg, August 1993.

[Pap94] Christos H Papadimitriou. On the complexity of the parity argument and other inefficient proofs of existence. *Journal of Computer and system Sciences*, 48(3):498–532, 1994.

[Pas13] Rafael Pass. Unprovable security of perfect nizk and non-interactive non-malleable commitments. In *Proceedings of the 10th theory of cryptography conference on Theory of Cryptography*, pages 334–354. Springer-Verlag, 2013.

[Pei16] Chris Peikert. A decade of lattice cryptography. *Foundations and Trends in Theoretical Computer Science*, 10(4):283–424, 2016.

[Pie18] Krzysztof Pietrzak. Simple Verifiable Delay Functions. In Avrim Blum, editor, *10th Innovations in Theoretical Computer Science Conference (ITCS 2019)*, volume 124 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 60:1–60:15, Dagstuhl, Germany, 2018. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[Pom87] Carl Pomerance. Fast, rigorous factorization and discrete logarithm algorithms. In *Discrete algorithms and complexity*, pages 119–143. Elsevier, 1987.

[PR17] Omer Paneth and Guy N. Rothblum. On zero-testable homomorphic encryption and publicly verifiable non-interactive arguments. In *TCC (2)*, volume 10678 of *Lecture Notes in Computer Science*, pages 283–315. Springer, 2017.

[PRSD17] Chris Peikert, Oded Regev, and Noah Stephens-Davidowitz. Pseudorandomness of ring-lwe for any ring and modulus. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, pages 461–473. ACM, 2017.

[PS96] David Pointcheval and Jacques Stern. Security proofs for signature schemes. In Ueli M. Maurer, editor, *EUROCRYPT'96*, volume 1070 of *LNCS*, pages 387–398. Springer, Heidelberg, May 1996.

[PS18] Chris Peikert and Sina Shiehian. Privately constraining and programming PRFs, the LWE way. In Michel Abdalla and Ricardo Dahab, editors, *PKC 2018, Part II*, volume 10770 of *LNCS*, pages 675–701. Springer, Heidelberg, March 2018.

[PS19] Chris Peikert and Sina Shiehian. Noninteractive zero knowledge for NP from (plain) learning with errors. In Alexandra Boldyreva and Daniele

Micciancio, editors, *CRYPTO 2019, Part I*, volume 11692 of *LNCS*, pages 89–114. Springer, Heidelberg, August 2019.

[PS20] Chris Peikert and Sina Shiehian. Constraining and watermarking PRFs from milder assumptions. In Aggelos Kiayias, Markulf Kohlweiss, Petros Wallden, and Vassilis Zikas, editors, *PKC 2020, Part I*, volume 12110 of *LNCS*, pages 431–461. Springer, Heidelberg, May 2020.

[PV05] Farzad Parvaresh and Alexander Vardy. Correcting errors beyond the Guruswami-Sudan radius in polynomial time. In *46th FOCS*, pages 285–294. IEEE Computer Society Press, October 2005.

[PVW08] Chris Peikert, Vinod Vaikuntanathan, and Brent Waters. A framework for efficient and composable oblivious transfer. In *Annual international cryptology conference*, pages 554–571. Springer, 2008.

[PW08] Chris Peikert and Brent Waters. Lossy trapdoor functions and their applications. In Richard E. Ladner and Cynthia Dwork, editors, *40th ACM STOC*, pages 187–196. ACM Press, May 2008.

[Ran38] Robert Alexander Rankin. The difference between consecutive prime numbers. *Journal of the London Mathematical Society*, 1(4):242–247, 1938.

[Reg04] Oded Regev. Lattices in computer science - average case hardness, 2004. Lecture Notes for Class (scribe: Elad Verbin). https://cims.nyu.edu/~regev/teaching/lattices_fall_2004/ln/averagecase.pdf.

[Reg05] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In Harold N. Gabow and Ronald Fagin, editors, *37th ACM STOC*, pages 84–93. ACM Press, May 2005.

[Riv99] Description of the lcs35 time capsule crypto-puzzle, 1999.

[Rog91] Phillip Rogaway. *The Round Complexity of secure Protocols*. PhD thesis, Massachusetts Institute of Technology, 1991.

[Rom90] John Rompel. One-way functions are necessary and sufficient for secure signatures. In *STOC*, pages 387–394. ACM, 1990.

[Rot13] Ron D Rothblum. On the circular security of bit-encryption. In *Theory of Cryptography*, pages 579–598. Springer, 2013.

[RRR16] Omer Reingold, Guy N. Rothblum, and Ron D. Rothblum. Constant-round interactive proofs for delegating computation. In Daniel Wichs and Yishay Mansour, editors, *48th ACM STOC*, pages 49–62. ACM Press, June 2016.

[RS09] Alon Rosen and Gil Segev. Chosen-ciphertext security via correlated products. In *Theory of Cryptography Conference*, pages 419–436. Springer, 2009.

[RSW96] Ronald L Rivest, Adi Shamir, and David A Wagner. Time-lock puzzles and timed-release crypto. 1996.

[RW18] Atri Rudra and Mary Wootters. Average-radius list-recoverability of random linear codes. In Artur Czumaj, editor, *29th SODA*, pages 644–662. ACM-SIAM, January 2018.

[SCG$^+$14] Eli Ben Sasson, Alessandro Chiesa, Christina Garman, Matthew Green, Ian Miers, Eran Tromer, and Madars Virza. Zerocash: Decentralized anonymous payments from bitcoin. In *2014 IEEE Symposium on Security and Privacy*, pages 459–474. IEEE, 2014.

[Sch87] Claus-Peter Schnorr. A hierarchy of polynomial time lattice basis reduction algorithms. *Theor. Comput. Sci.*, 53:201–224, 1987.

[Sch89] Claus-Peter Schnorr. Efficient identification and signatures for smart cards. In *CRYPTO 1989*, pages 239–252. Springer, 1989.

[SE94] Claus-Peter Schnorr and Martin Euchner. Lattice basis reduction: Improved practical algorithms and solving subset sum problems. *Mathematical programming*, 66(1-3):181–199, 1994.

[Sha90] Adi Shamir. IP=PSPACE. In *31st FOCS*, pages 11–15. IEEE Computer Society Press, October 1990.

[Sho97] Victor Shoup. Lower bounds for discrete logarithms and related problems. In *International Conference on the Theory and Applications of Cryptographic Techniques*, pages 256–266. Springer, 1997.

[Sim98] Daniel R. Simon. Finding collisions on a one-way street: Can secure hash functions be based on general assumptions? In Kaisa Nyberg, editor, *EUROCRYPT'98*, volume 1403 of *LNCS*, pages 334–345. Springer, Heidelberg, May / June 1998.

[SS94] Michael Sipser and Daniel A. Spielman. Expander codes. In *35th FOCS*, pages 566–576. IEEE Computer Society Press, November 1994.

[Sta] Stanford Center for Blockchain Research. The Stanford center for blockchain research. https://cbr.stanford.edu/.

[SW14] Amit Sahai and Brent Waters. How to use indistinguishability obfuscation: deniable encryption, and more. In David B. Shmoys, editor, *46th ACM STOC*, pages 475–484. ACM Press, May / June 2014.

[Vad12] Salil P. Vadhan. *Pseudorandomness*. Now Publishers Inc., 2012. https://people.seas.harvard.edu/~salil/pseudorandomness/.

[Val08] Paul Valiant. Incrementally verifiable computation or proofs of knowledge imply time/space efficiency. In Ran Canetti, editor, *TCC 2008*, volume 4948 of *LNCS*, pages 1–18. Springer, Heidelberg, March 2008.

[W$^+$14] Gavin Wood et al. Ethereum: A secure decentralised generalised transaction ledger. *Ethereum project yellow paper*, 151(2014):1–32, 2014.

[Wee05] Hoeteck Wee. On obfuscating point functions. In *Proceedings of the thirty-seventh annual ACM symposium on Theory of computing*, pages 523–532. ACM, 2005.

[Wes19] Benjamin Wesolowski. Efficient verifiable delay functions. In *EUROCRYPT 2019*, pages 379–407. Springer, 2019.

[Wic18] Daniel Wichs. personal communication, April 2018.

[WTs$^+$18] Riad S. Wahby, Ioanna Tzialla, abhi shelat, Justin Thaler, and Michael Walfish. Doubly-efficient zkSNARKs without trusted setup. In *2018 IEEE Symposium on Security and Privacy*, pages 926–943. IEEE Computer Society Press, May 2018.

[WW21] Hoeteck Wee and Daniel Wichs. Candidate obfuscation via oblivious LWE sampling. In Anne Canteaut and François-Xavier Standaert, editors, *EUROCRYPT 2021, Part III*, volume 12698 of *LNCS*, pages 127–156. Springer, Heidelberg, October 2021.

[WZ17] Daniel Wichs and Giorgos Zirdelis. Obfuscating compute-and-compare programs under lwe. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 600–611. IEEE, 2017.

[Yao86] Andrew Chi-Chih Yao. How to generate and exchange secrets. In *27th Annual Symposium on Foundations of Computer Science (sfcs 1986)*, pages 162–167. IEEE, 1986.

[YD17] Yang Yu and Léo Ducas. Second order statistical behavior of LLL and BKZ. In *International Conference on Selected Areas in Cryptography*, pages 3–22. Springer, 2017.

[YZW$^+$17] Yu Yu, Jiang Zhang, Jian Weng, Chun Guo, and Xiangxue Li. Learning parity with noise implies collision resistant hashing. 2017. https://eprint.iacr.org/2017/1260.pdf.

[Zha16] Mark Zhandry. The magic of ELFs. In Matthew Robshaw and Jonathan Katz, editors, *CRYPTO 2016, Part I*, volume 9814 of *LNCS*, pages 479–508. Springer, Heidelberg, August 2016.

[ZKP] ZKProof. Zero-knowledge proof standardization. https://zkproof.org/.