Fiat–Shamir via List-Recoverable Codes (or: Parallel Repetition of GMW is not Zero-Knowledge)*

Justin Holmgren
NTT Research
USA
justin.holmgren@ntt-research.com

Alex Lombardi
MIT
USA
alexjl@mit.edu

Ron D. Rothblum
Technion
Israel
rothblum@cs.technion.ac.il

ABSTRACT

In a seminal work, Goldreich, Micali and Wigderson (CRYPTO '86) demonstrated the wide applicability of zero-knowledge proofs by constructing such a proof system for the NP-complete problem of graph 3-coloring. A long-standing open question has been whether parallel repetition of their protocol preserves zero knowledge. In this work, we answer this question in the negative, assuming a standard cryptographic assumption (i.e., the hardness of learning with errors (LWE)).

Leveraging a connection observed by Dwork, Naor, Reingold, and Stockmeyer (FOCS '99), our negative result is obtained by making *positive* progress on a related fundamental problem in cryptography: securely instantiating the Fiat–Shamir heuristic for eliminating interaction in public-coin interactive protocols. A recent line of work has shown how to instantiate the heuristic securely, albeit only for a limited class of protocols.

Our main result shows how to instantiate Fiat–Shamir for parallel repetitions of much more general interactive proofs. In particular, we construct hash functions that, assuming LWE, securely realize the Fiat–Shamir transform for the following rich classes of protocols:

1) The parallel repetition of any "commit-and-open" protocol (such as the GMW protocol mentioned above), when a specific (natural) commitment scheme is used. Commit-and-open protocols are a ubiquitous paradigm for constructing general purpose public-coin zero knowledge proofs.

2) The parallel repetition of any base protocol that (1) satisfies a stronger notion of soundness called round-by-round soundness, and (2) has an efficient procedure, using a suitable trapdoor, for recognizing "bad verifier randomness" that would allow the prover to cheat.

Our results are obtained by establishing a new connection between the Fiat–Shamir transform and *list-recoverable codes*. In contrast to the usual focus in coding theory, we focus on a parameter regime in which the input lists are extremely large, but the rate can be small. We give a (probabilistic) construction based on Parvaresh-Vardy codes (FOCS '05) that suffices for our applications.

*The full version of this paper is available at [47].

STOC '21, June 21–25, 2021, Virtual, Italy

ACM ISBN 978-1-4503-8053-9/21/06.
https://doi.org/10.1145/3406325.3451116

CCS CONCEPTS

• **Theory of computation → Interactive proof systems**; *Cryptographic protocols*; *Cryptographic primitives.*

KEYWORDS

Fiat–Shamir heuristic, list-recoverable codes, cryptographic protocols, zero-knowledge protocols

ACM Reference Format:
Justin Holmgren, Alex Lombardi, and Ron D. Rothblum. 2021. Fiat–Shamir via List-Recoverable Codes (or: Parallel Repetition of GMW is not Zero-Knowledge). In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing (STOC '21), June 21–25, 2021, Virtual, Italy*. ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3406325.3451116

1 INTRODUCTION

Zero-knowledge proofs, introduced by Goldwasser, Micali and Rackoff [35], 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 [31] 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 [32]), 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. (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.)

Curiously, it has long been known that parallel repetition does not preserve zero-knowledge for some (contrived) protocols [30]. However, for "naturally occurring" protocols, the question remained open for decades. A sequence of recent works [15, 18, 46, 52] 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 [26]), 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.* [16] and Peikert and Shiehian [67], who showed that some *classical* ZK protocols [12, 35] 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 [5, 23]:

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) [69].

Theorem 1.1 (Informally Stated). *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 1.1:

- The commitment scheme C used in order to prove Theorem 1.1 is a natural one. (In fact, this instantiation dates back to the original [31] paper.) 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 1.1 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 the full version [47] for further discussion).

We prove Theorem 1.1 through a more general result showing that parallel repetition does not preserve zero-knowledge for a large class of protocols. This class includes all general-purpose public-coin (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.) 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 [50] and more generally based on zero-knowledge PCPs (see, e.g., a recent survey [49]).

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.

1.1 Securely Instantiating Fiat–Shamir

The Fiat–Shamir heuristic [25] is a generic technique for eliminating interaction in *public-coin* interactive proofs. (The original goal in [25] was to efficiently compile (interactive) identification schemes into signature schemes, but the technique is applicable to more general protocols.) 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* [8, 9, 68]. In reality however, we need to realize the hash function with a concrete cryptographic hash function. Following [16], we say that a hash function family $\mathcal{H}$ is FS-compatible (We remark that the term "FS-compatible" has a different meaning in a recent work of [51]. More specifically, [51] 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.) 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 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 [23] 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. (Roughly speaking, [23] 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.) This means, in particular, that in order to prove Theorem 1.1, 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 [3, 10, 19, 33]. However, a recent line of work [14, 16, 18, 46, 51, 52, 57, 67] 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 [16, 67] 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 [12] 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 **NP**-complete, the works of [16, 67] yielded *non-interactive* zero-knowledge (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.) 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 [16, 67] 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 [16, 67] 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.

Theorem 1.2 (Informally Stated). *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 [23], Theorem 1.1 follows immediately from Theorem 1.2.

Remark 1.3. *An important example of a commit-and-open protocol is Kilian's [53] celebrated succinct argument-system, as well as its generalizations based on interactive oracle proofs [9]. However, we point out that Theorem 1.2 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 [63]), remains a fundamental open problem (see also [6, 28]).

Because it applies to parallel repetitions of *all* commit-and-open protocols (rather than just those with a single bad challenge), Theorem 1.2 substantially generalizes the class of protocols that have sound Fiat–Shamir instantiations in the standard model. We believe that Theorem 1.2 (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 [9] for more details).

Thus, following [16], 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 {accept, 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* [60] and the related [34] protocol (see [16, 51] for details).

We say that a protocol with RBR soundness has 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 [16, 67] 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 1.4 (Informally Stated). *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 1.5. *Theorem 1.2 actually follows from Theorem 1.4 since constant-round protocols with negligible soundness error 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 1.2 is simpler than that of Theorem 1.4 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 1.4 as we did for Theorem 1.2.

Finally, we note that Theorem 1.4 can be combined with the main insight of [51] (which is orthogonal to our work) to *further* generalize the class of protocols Π that have sound Fiat–Shamir instantiations. Informally, the [51] 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 [34] protocol, even when the field size of the base protocol is *poly-logarithmic*. In contrast, [51] can only handle variants of [34] with an exponential field size. ([51] use a large field in order to have negligible soundness error but only polynomially many bad challenges.) For example, this precludes applications in which one needs to materialize entire truth tables of polynomials over the field.

1.2 Technical Overview

We now describe our techniques for proving Theorem 1.2, with a particular focus on the GMW protocol for ease of understanding. Our starting point is the work of [16], which gave the first instantiation of Fiat–Shamir in the standard model based on standard cryptographic assumptions. As in prior work [18, 46, 52], their Fiat–Shamir instantiation makes use of the framework of *correlation intractability* [19], which we recall here. (In fact, [23] cites personal communication with Chaum and Impagliazzo for an early variant of this connection. Full formalizations of this paradigm appear in [16, 18].)

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 [2, 14, 16–18, 36, 46, 51, 52, 55–59, 67]) 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 1.4, but for now, we recall the following result from [67], 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 1.6 ([67], 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.*

As described in the theorem statement, Theorem 1.6 has the following two limitations (which are also present in the predecessor work [16] (More specifically, this limitation is present in the subset of results in [16] that are based on quantitatively standard cryptographic assumptions)).

- 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.

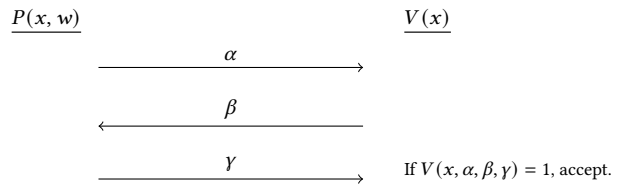

Figure 1: A 3-message public coin interactive proof Π.

After applying the Fiat–Shamir transform using hash family $\mathcal{H}$, we obtain the protocol $\Pi_{\mathrm{FS},\mathcal{H}}$ below.

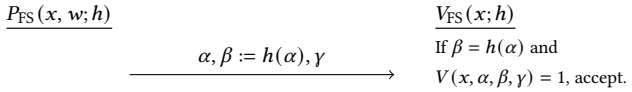

Figure 2: The Protocol $\Pi_{\mathrm{FS},\mathcal{H}}$.

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:

$$R^{(0)}_{x,\Pi} = \{(\alpha, \beta) : \exists \gamma \text{ s.t. } V(x, \alpha, \beta, \gamma) = 1\}.$$

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 1.6, 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.

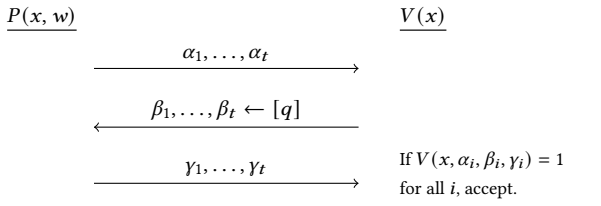

Figure 3: A parallel-repeated protocol Π^t.

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 (A decommitment (m, r) of a string com is a message m and choice of commitment randomness r such that $\mathrm{com} = \mathsf{Com}(m; r)$.) (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 $q = |E|$. (Our results in this overview may appear to require that q is polynomial in n, but we show in the full version [47] 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.)

Recall that by Theorem 1.6, 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. As a first step, we show (following [16, 46]) 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

$$R_{x,\Pi^t} := \Big\{\big((\alpha_1, \ldots, \alpha_t), (\beta_1, \ldots, \beta_t)\big) : \\ \text{each } z_i := \mathsf{Extract}(\alpha_i[\beta_i]) \text{ has two distinct colors}\Big\},$$

where 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 [16, 67] results, namely, that R is *not* a function. We quantify the extent to which R is not a function with the following terminology.

Definition 1.7 (d-Bounded Relation). *We say that a relation $R \subseteq \{0,1\}^n \times \{0,1\}^m$ is $d = d(n)$-*bounded *if $|R(x)| \le d$, 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 [16, 67] were shown to satisfy correlation intractability for (efficiently computable) *functions*, i.e., 1-bounded relations. In prior work [16, 51], 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". (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 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.

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 [16, 67]).

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* (The type of derandomization that we require is related to, but different from, the "sampler-based" [29, 72] derandomized parallel repetition of Bellare, Goldreich and Goldwasser [7]. The exact approach of [7] does not work for us for reasons similar to the "naive" PRG approach below.) 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 [13] or "Nisan-Wigderson style" [1, 48, 65]; 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 and first messages $\alpha_1, \ldots, \alpha_t$, the probability that $G(r) = (\beta_1, \ldots, \beta_t)$ 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, (This boils down to the sub-optimal 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).) 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:

$$\tilde{R}_{x,\Pi,G} = \left\{(\alpha, r) : (\alpha, G(r)) \in R_{x,\Pi^t}\right\},$$

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 $\mathsf{Samp}(r) \to (\beta_1, \ldots, \beta_t)$ along with a hash family $\mathcal{H}$ that is correlation intractable for the relation $\tilde{R}$, which can be expressed as

$$\tilde{R}_{x,\Pi^t,\mathsf{Samp}} = \left\{(\alpha, r) : (\alpha_i, \mathsf{Samp}(r)_i) \in R_{x,\Pi} \text{ for all } i\right\}.$$

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 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

$$\tilde{R}_{x,\Pi^t,\mathsf{Samp}} = \left\{(\alpha, r) : \mathsf{Samp}(r)_i \in S_i \text{ for all } i\right\},$$

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 [16, 67] – 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 [37]!

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 (Encode, 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 output list).

List-recoverable codes were introduced by [37] 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. (The dependence is actually allowed to be $\mathsf{poly}(\lambda, q, 1/\epsilon)$)

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 [45, Figure 1] as well as [70] for examples), the input list size ℓ is typically very small (For example, degree k Reed-Solomon codes over $\mathbf{F}_q$ can handle $\ell \leq \frac{q}{k}$, while known higher rate constructions can only tolerate much smaller values of ℓ.) 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 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 1.8 (Informally stated). *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*
- (Encode, 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 the full version of this paper, we rephrase Theorem 1.8 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. We then show how this form of CI allows us to prove our general FS results: Theorem 1.2 and Theorem 1.4. 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 $1/\epsilon$; 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 [41, 42, 66], expander codes [45, 71], and codes built by a combination of these tools [37–40]. 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 $\ell = q^{o(1)}$; (An interesting concurrent and independent work [22] 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.) 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* [27] 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 [37] (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 [37].

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 $y = \mathrm{Encode}_{\mathrm{out}}(m)$ 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 $C_{\mathrm{out}}, C_{\mathrm{in}}$ that meet our demands.

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 $C_{\mathrm{out}} \circ C_{\mathrm{in}}$ 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 $C_{\mathrm{out}} \circ C_{\mathrm{in}}$ 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 1.9 (Informal). *For all $\ell < q = \mathrm{poly}(\lambda)$, 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 $(\ell, \mathrm{poly}(\lambda))$-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 the full version [47] for more details.

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 $S = S_1 \times \ldots S_t \subseteq [q]^t$ (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 [16, 67], 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 [67] 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 [42, 66] 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.

1.4 Related Work

1.4.1 Correlation Intractability and Fiat–Shamir. We survey the recent constructions of correlation intractable (CI) hash families [14, 16–18, 46, 52, 67] 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 [15, 17, 18, 46, 52] 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,

- [17] 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 [4].
- [18, 52] 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 [28, 64]. For example, [18] 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. [52] makes a simiar assumption, and additionally assumes (subexponentially secure) iO.
- [46] constructs a hash family that is CI for all "efficiently sampleable relations" (similar in spirit but technically incomparable to "efficiently verifiable relations" as in [17]) 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. [16] (see [15]) 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 [16] (see [20]), a sequence of works [14, 16, 67] gave constructions of restricted forms of correlation intractability based on widely accepted assumptions. In more detail,

- [16, 67] 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. [16] constructs such a hash family under circular-secure

fully homomorphic encryption, while [67] relies on the plain LWE assumption.
- [14] 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 $\mathbf{F}_2$, 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. (Indeed, the constructions of [16, 67] also support a kind of error tolerance, although this was irrelevant for their purposes.) 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*. (For 3-message protocols, these are abstracted as "trapdoor Σ-protocols" in [16].) 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 the form (α, β, γ) exists. Proof systems satisfying this notion include important protocols such as [12, 24, 35], 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 [51, 57], 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 [14, 16, 67], 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 [31], 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 [16, 67] 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 [16, 67] 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 [16, 67] 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, 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 [14]. 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.

1.4.2 *List-Recoverable Codes and Cryptography.*

List-recoverable codes have previously been used [11, 21, 44, 54, 61] in cryptography in the context of *domain extension* [62] 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 [44] 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 [21, 61]), it suffices to use off-the-shelf Parvaresh-Vardy codes [43], albeit in somewhat non-standard parameter regimes. For example, [44] 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 [11, 21, 44, 54, 61] 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).

ACKNOWLEDGEMENTS

We thank Vinod Vaikuntanathan for helpful discussions and feedback.

AL conducted research in part while he was an intern at NTT Research. AL was supported in part by an NDSEG fellowship, by NSF Grants CNS-1350619 and CNS-1414119, and by the Defense Advanced Research Projects Agency (DARPA) and the U.S. Army Research Office under contracts W911NF-15-C-0226 and W911NF-15-C-0236.

RR was supported in part by a Milgrom family grant, by the Israeli Science Foundation (Grants No. 1262/18 and 2137/19), and grants from the Technion Hiroshi Fujiwara cyber security research center and Israel cyber directorate.

This material is based upon work supported by DARPA (for the second author) under Agreement No. HR00112020023. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the United States Government or DARPA.

REFERENCES

[1] Vikraman Arvind and Johannes Köbler. 1997. On resource-bounded measure and pseudorandomness. In *International Conference on Foundations of Software Technology and Theoretical Computer Science*. Springer, 235–249.

[2] Saikrishna Badrinarayanan, Rex Fernando, Aayush Jain, Dakshita Khurana, and Amit Sahai. 2020. Statistical ZAP Arguments. In *EUROCRYPT 2020, Part III (LNCS, Vol. 12107)*, Anne Canteaut and Yuval Ishai (Eds.). Springer, Heidelberg, 642–667. https://doi.org/10.1007/978-3-030-45727-3_22

[3] Boaz Barak. 2001. How to Go Beyond the Black-Box Simulation Barrier. In *42nd FOCS*. IEEE Computer Society Press, 106–115. https://doi.org/10.1109/SFCS.2001.959885

[4] Boaz Barak, Nir Bitansky, Ran Canetti, Yael Tauman Kalai, Omer Paneth, and Amit Sahai. 2014. Obfuscation for Evasive Functions. In *TCC 2014 (LNCS, Vol. 8349)*, Yehuda Lindell (Ed.). Springer, Heidelberg, 26–51. https://doi.org/10.1007/978-3-642-54242-8_2

[5] Boaz Barak, Yehuda Lindell, and Salil P. Vadhan. 2003. Lower Bounds for Non-Black-Box Zero Knowledge. In *44th FOCS*. IEEE Computer Society Press, 384–393. https://doi.org/10.1109/SFCS.2003.1238212

[6] James Bartusek, Liron Bronfman, Justin Holmgren, Fermi Ma, and Ron D. Rothblum. 2019. On the (In)security of Kilian-Based SNARGs. In *TCC 2019, Part II (LNCS, Vol. 11892)*, Dennis Hofheinz and Alon Rosen (Eds.). Springer, Heidelberg, 522–551. https://doi.org/10.1007/978-3-030-36033-7_20

[7] Mihir Bellare, Oded Goldreich, and Shafi Goldwasser. 1990. Randomness in Interactive Proofs. In *31st FOCS*. IEEE Computer Society Press, 563–572. https://doi.org/10.1109/FSCS.1990.89577

[8] Mihir Bellare and Phillip Rogaway. 1994. Entity Authentication and Key Distribution. In *CRYPTO'93 (LNCS, Vol. 773)*, Douglas R. Stinson (Ed.). Springer, Heidelberg, 232–249. https://doi.org/10.1007/3-540-48329-2_21

[9] Eli Ben-Sasson, Alessandro Chiesa, and Nicholas Spooner. 2016. Interactive Oracle Proofs. In *TCC 2016-B, Part II (LNCS, Vol. 9986)*, Martin Hirt and Adam D. Smith (Eds.). Springer, Heidelberg, 31–60. https://doi.org/10.1007/978-3-662-53644-5_2

[10] Nir Bitansky, Dana Dachman-Soled, Sanjam Garg, Abhishek Jain, Yael Tauman Kalai, Adriana López-Alt, and Daniel Wichs. 2013. Why "Fiat-Shamir for Proofs" Lacks a Proof. In *TCC 2013 (LNCS, Vol. 7785)*, Amit Sahai (Ed.). Springer, Heidelberg, 182–201. https://doi.org/10.1007/978-3-642-36594-2_11

[11] Nir Bitansky, Yael Tauman Kalai, and Omer Paneth. 2018. Multi-collision resistance: a paradigm for keyless hash functions. In *50th ACM STOC*, Ilias Diakonikolas, David Kempe, and Monika Henzinger (Eds.). ACM Press, 671–684. https://doi.org/10.1145/3188745.3188870

[12] Manuel Blum. 1986. How to prove a theorem so no one else can claim it. In *Proceedings of the International Congress of Mathematicians*, Vol. 1. Citeseer, 2.

[13] Manuel Blum and Silvio Micali. 1982. How to Generate Cryptographically Strong Sequences of Pseudo Random Bits. In *23rd FOCS*. IEEE Computer Society Press, 112–117. https://doi.org/10.1109/SFCS.1982.72

[14] Zvika Brakerski, Venkata Koppula, and Tamer Mour. 2020. NIZK from LPN and Trapdoor Hash via Correlation Intractability for Approximable Relations. In *CRYPTO 2020, Part III (LNCS, Vol. 12172)*, Daniele Micciancio and Thomas Ristenpart (Eds.). Springer, Heidelberg, 738–767. https://doi.org/10.1007/978-3-030-56877-1_26

[15] Ran Canetti, Yilei Chen, Justin Holmgren, Alex Lombardi, Guy N. Rothblum, and Ron D. Rothblum. 2018. Fiat-Shamir From Simpler Assumptions. Cryptology ePrint Archive, Report 2018/1004. https://eprint.iacr.org/2018/1004. Part 1 of [16].

[16] Ran Canetti, Yilei Chen, Justin Holmgren, Alex Lombardi, Guy N. Rothblum, Ron D. Rothblum, and Daniel Wichs. 2019. Fiat-Shamir: from practice to theory. In *51st ACM STOC*, Moses Charikar and Edith Cohen (Eds.). ACM Press, 1082–1090. https://doi.org/10.1145/3313276.3316380

[17] Ran Canetti, Yilei Chen, and Leonid Reyzin. 2016. On the Correlation Intractability of Obfuscated Pseudorandom Functions. In *TCC 2016-A, Part I (LNCS, Vol. 9562)*, Eyal Kushilevitz and Tal Malkin (Eds.). Springer, Heidelberg, 389–415. https://doi.org/10.1007/978-3-662-49096-9_17

[18] Ran Canetti, Yilei Chen, Leonid Reyzin, and Ron D. Rothblum. 2018. Fiat-Shamir and Correlation Intractability from Strong KDM-Secure Encryption. In *EUROCRYPT 2018, Part I (LNCS, Vol. 10820)*, Jesper Buus Nielsen and Vincent Rijmen (Eds.). Springer, Heidelberg, 91–122. https://doi.org/10.1007/978-3-319-78381-9_4

[19] Ran Canetti, Oded Goldreich, and Shai Halevi. 1998. The Random Oracle Methodology, Revisited (Preliminary Version). In *30th ACM STOC*. ACM Press, 209–218. https://doi.org/10.1145/276698.276741

[20] Ran Canetti, Alex Lombardi, and Daniel Wichs. 2018. Fiat-Shamir: From Practice to Theory, Part II (NIZK and Correlation Intractability from Circular-Secure FHE). Cryptology ePrint Archive, Report 2018/1248. https://eprint.iacr.org/2018/1248. Part 2 of [16].

[21] Yevgeniy Dodis and John P. Steinberger. 2011. Domain Extension for MACs Beyond the Birthday Barrier. In *EUROCRYPT 2011 (LNCS, Vol. 6632)*, Kenneth G. Paterson (Ed.). Springer, Heidelberg, 323–342. https://doi.org/10.1007/978-3-642-20465-4_19

[22] Dean Doron and Mary Wootters. 2020. High-Probability List-Recovery, and Applications to Heavy Hitters. *ECCC* (2020). https://eccc.weizmann.ac.il/report/2020/162/.

[23] Cynthia Dwork, Moni Naor, Omer Reingold, and Larry J. Stockmeyer. 1999. Magic Functions. In *40th FOCS*. IEEE Computer Society Press, 523–534. https://doi.org/10.1109/SFFCS.1999.814626

[24] Uriel Feige, Dror Lapidot, and Adi Shamir. 1990. Multiple Non-Interactive Zero Knowledge Proofs Based on a Single Random String (Extended Abstract). In *31st FOCS*. IEEE Computer Society Press, 308–317. https://doi.org/10.1109/FSCS.1990.89549

[25] Amos Fiat and Adi Shamir. 1986. How to Prove Yourself: Practical Solutions to Identification and Signature Problems. In *CRYPTO'86 (LNCS, Vol. 263)*, Andrew M. Odlyzko (Ed.). Springer, Heidelberg, 186–194. https://doi.org/10.1007/3-540-47721-7_12

[26] Nils Fleischhacker, Vipul Goyal, and Abhishek Jain. 2018. On the Existence of Three Round Zero-Knowledge Proofs. In *EUROCRYPT 2018, Part III (LNCS, Vol. 10822)*, Jesper Buus Nielsen and Vincent Rijmen (Eds.). Springer, Heidelberg, 3–33. https://doi.org/10.1007/978-3-319-78372-7_1

[27] G David Forney. 1966. Concatenated codes. (1966).

[28] Craig Gentry and Daniel Wichs. 2011. Separating succinct non-interactive arguments from all falsifiable assumptions. In *43rd ACM STOC*, Lance Fortnow and Salil P. Vadhan (Eds.). ACM Press, 99–108. https://doi.org/10.1145/1993636.1993651

[29] Oded Goldreich. 2011. A sample of samplers: A computational perspective on sampling. In *Studies in Complexity and Cryptography. Miscellanea on the Interplay between Randomness and Computation*. Springer, 302–332.

[30] Oded Goldreich and Hugo Krawczyk. 1996. On the composition of zero-knowledge proof systems. *SIAM J. Comput.* 25, 1 (1996), 169–192.

[31] Oded Goldreich, Silvio Micali, and Avi Wigderson. 1986. How to Prove all NP-Statements in Zero-Knowledge, and a Methodology of Cryptographic Protocol Design. In *CRYPTO'86 (LNCS, Vol. 263)*, Andrew M. Odlyzko (Ed.). Springer, Heidelberg, 171–185. https://doi.org/10.1007/3-540-47721-7_11

[32] Oded Goldreich and Yair Oren. 1994. Definitions and Properties of Zero-Knowledge Proof Systems. *Journal of Cryptology* 7, 1 (Dec. 1994), 1–32. https://doi.org/10.1007/BF00195207

[33] Shafi Goldwasser and Yael Tauman Kalai. 2003. On the (In)security of the Fiat-Shamir Paradigm. In *44th FOCS*. IEEE Computer Society Press, 102–115. https://doi.org/10.1109/SFCS.2003.1238185

[34] Shafi Goldwasser, Yael Tauman Kalai, and Guy N. Rothblum. 2008. Delegating computation: interactive proofs for muggles. In *40th ACM STOC*, Richard E. Ladner and Cynthia Dwork (Eds.). ACM Press, 113–122. https://doi.org/10.1145/1374376.1374396

[35] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. 1985. The Knowledge Complexity of Interactive Proof-Systems (Extended Abstract). In *17th ACM STOC*. ACM Press, 291–304. https://doi.org/10.1145/22145.22178

[36] Vipul Goyal, Abhishek Jain, Zhengzhong Jin, and Giulio Malavolta. 2020. Statistical Zaps and New Oblivious Transfer Protocols. In *EUROCRYPT 2020, Part III (LNCS, Vol. 12107)*, Anne Canteaut and Yuval Ishai (Eds.). Springer, Heidelberg, 668–699. https://doi.org/10.1007/978-3-030-45727-3_23

[37] Venkatesan Guruswami and Piotr Indyk. 2001. Expander-Based Constructions of Efficiently Decodable Codes. In *42nd FOCS*. IEEE Computer Society Press, 658–667. https://doi.org/10.1109/SFCS.2001.959942

[38] Venkatesan Guruswami and Piotr Indyk. 2002. Near-optimal linear-time codes for unique decoding and new list-decodable codes over smaller alphabets. In *34th ACM STOC*. ACM Press, 812–821. https://doi.org/10.1145/509907.510023

[39] Venkatesan Guruswami and Piotr Indyk. 2003. Linear time encodable and list decodable codes. In *35th ACM STOC*. ACM Press, 126–135. https://doi.org/10.1145/780542.780562

[40] Venkatesan Guruswami and Piotr Indyk. 2004. Linear-Time List Decoding in Error-Free Settings: (Extended Abstract). In *ICALP 2004 (LNCS, Vol. 3142)*, Josep Díaz, Juhani Karhumäki, Arto Lepistö, and Donald Sannella (Eds.). Springer, Heidelberg, 695–707. https://doi.org/10.1007/978-3-540-27836-8_59

[41] Venkatesan Guruswami and Atri Rudra. 2008. Soft decoding, dual bch codes, and better list-decodable e-biased codes. In *2008 23rd Annual IEEE Conference on Computational Complexity*. IEEE, 163–174.

[42] Venkatesan Guruswami and Madhu Sudan. 1998. Improved Decoding of Reed-Solomon and Algebraic-Geometric Codes. In *39th FOCS*. IEEE Computer Society Press, 28–39. https://doi.org/10.1109/SFCS.1998.743426

[43] Venkatesan Guruswami, Christopher Umans, and Salil Vadhan. 2009. Unbalanced expanders and randomness extractors from Parvaresh–Vardy codes. *Journal of the ACM (JACM)* 56, 4 (2009), 1–34.

[44] Iftach Haitner, Yuval Ishai, Eran Omri, and Ronen Shaltiel. 2015. Parallel Hashing via List Recoverability. In *CRYPTO 2015, Part II (LNCS, Vol. 9216)*, Rosario Gennaro and Matthew J. B. Robshaw (Eds.). Springer, Heidelberg, 173–190. https://doi.org/10.1007/978-3-662-48000-7_9

[45] Brett Hemenway and Mary Wootters. 2015. Linear-Time List Recovery of High-Rate Expander Codes. In *ICALP 2015, Part I (LNCS, Vol. 9134)*, Magnús M. Halldórsson, Kazuo Iwama, Naoki Kobayashi, and Bettina Speckmann (Eds.). Springer, Heidelberg, 701–712. https://doi.org/10.1007/978-3-662-47672-7_57

[46] Justin Holmgren and Alex Lombardi. 2018. Cryptographic Hashing from Strong One-Way Functions (Or: One-Way Product Functions and Their Applications). In *59th FOCS*, Mikkel Thorup (Ed.). IEEE Computer Society Press, 850–858. https://doi.org/10.1109/FOCS.2018.00085

[47] Justin Holmgren, Alex Lombardi, and Ron D Rothblum. 2021. Fiat-Shamir via List-Recoverable Codes (or: Parallel Repetition of GMW is not Zero-Knowledge). *IACR Cryptology ePrint Archive* Report 2021/286 (2021).

[48] Russell Impagliazzo and Avi Wigderson. 1997. P = BPP if E Requires Exponential Circuits: Derandomizing the XOR Lemma. In *29th ACM STOC*. ACM Press, 220–229. https://doi.org/10.1145/258533.258590

[49] Yuval Ishai. 2020. Zero-Knowledge Proofs from Information-Theoretic Proof Systems. (2020). https://zkproof.org/2020/08/12/information-theoretic-proof-systems/.

[50] Yuval Ishai, Eyal Kushilevitz, Rafail Ostrovsky, and Amit Sahai. 2007. Zero-knowledge from secure multiparty computation. In *39th ACM STOC*, David S. Johnson and Uriel Feige (Eds.). ACM Press, 21–30. https://doi.org/10.1145/1250790.1250794

[51] Ruta Jawale, Yael Tauman Kalai, Dakshita Khurana, and Rachel Zhang. 2020. SNARGs for Bounded Depth Computations and PPAD Hardness from Sub-Exponential LWE. *IACR Cryptol. ePrint Arch* 2020 (2020), 980. To appear in STOC 2021.

[52] Yael Tauman Kalai, Guy N. Rothblum, and Ron D. Rothblum. 2017. From Obfuscation to the Security of Fiat-Shamir for Proofs. In *CRYPTO 2017, Part II (LNCS, Vol. 10402)*, Jonathan Katz and Hovav Shacham (Eds.). Springer, Heidelberg, 224–251. https://doi.org/10.1007/978-3-319-63715-0_8

[53] Joe Kilian. 1992. A Note on Efficient Zero-Knowledge Proofs and Arguments (Extended Abstract). In *24th ACM STOC*. ACM Press, 723–732. https://doi.org/10.1145/129712.129782

[54] Ilan Komargodski, Moni Naor, and Eylon Yogev. 2018. Collision Resistant Hashing for Paranoids: Dealing with Multiple Collisions. In *EUROCRYPT 2018, Part II (LNCS, Vol. 10821)*, Jesper Buus Nielsen and Vincent Rijmen (Eds.). Springer, Heidelberg, 162–194. https://doi.org/10.1007/978-3-319-78375-8_6

[55] Benoît Libert, Khoa Nguyen, Alain Passelègue, and Radu Titiu. 2019. Simulation-Sound Arguments for LWE and Applications to KDM-CCA2 Security. Cryptology ePrint Archive, Report 2019/908. https://eprint.iacr.org/2019/908.

[56] Benoît Libert, Khoa Nguyen, Thomas Peters, and Moti Yung. 2020. One-Shot Fiat-Shamir-based NIZK Arguments of Composite Residuosity in the Standard Model. Cryptology ePrint Archive, Report 2020/1334. https://eprint.iacr.org/2020/1334.

[57] Alex Lombardi and Vinod Vaikuntanathan. 2020. Fiat-Shamir for Repeated Squaring with Applications to PPAD-Hardness and VDFs. In *CRYPTO 2020, Part III (LNCS, Vol. 12172)*, Daniele Micciancio and Thomas Ristenpart (Eds.). Springer, Heidelberg, 632–651. https://doi.org/10.1007/978-3-030-56877-1_22

[58] Alex Lombardi and Vinod Vaikuntanathan. 2020. Multi-Input Correlation Intractable Hash Functions via Shift-Hiding. *IACR Cryptology ePrint Archive* Report 2020/1378 (2020). https://eprint.iacr.org/2020/1378.

[59] Alex Lombardi, Vinod Vaikuntanathan, and Daniel Wichs. 2019. 2-Message Publicly Verifiable WI from (Subexponential) LWE. Cryptology ePrint Archive, Report 2019/808. https://eprint.iacr.org/2019/808.

[60] C Lund, L Fortnow, H Karloff, and N Nisan. 1990. The polynomial-time hierarchy has interactive proofs. *Proceedings of STOC 1990* (1990), 2–10.

[61] Ueli M. Maurer and Stefano Tessaro. 2007. Domain Extension of Public Random Functions: Beyond the Birthday Barrier. In *CRYPTO 2007 (LNCS, Vol. 4622)*, Alfred Menezes (Ed.). Springer, Heidelberg, 187–204. https://doi.org/10.1007/978-3-540-74143-5_11

[62] Ralph C. Merkle. 1988. A Digital Signature Based on a Conventional Encryption Function. In *CRYPTO'87 (LNCS, Vol. 293)*, Carl Pomerance (Ed.). Springer, Heidelberg, 369–378. https://doi.org/10.1007/3-540-48184-2_32

[63] Silvio Micali. 1993. Fair Public-Key Cryptosystems. In *CRYPTO'92 (LNCS, Vol. 740)*, Ernest F. Brickell (Ed.). Springer, Heidelberg, 113–138. https://doi.org/10.1007/3-540-48071-4_9

[64] Moni Naor. 2003. On Cryptographic Assumptions and Challenges (Invited Talk). In *CRYPTO 2003 (LNCS, Vol. 2729)*, Dan Boneh (Ed.). Springer, Heidelberg, 96–109. https://doi.org/10.1007/978-3-540-45146-4_6

[65] Noam Nisan and Avi Wigderson. 1988. Hardness vs. Randomness (Extended Abstract). In *29th FOCS*. IEEE Computer Society Press, 2–11. https://doi.org/10.1109/SFCS.1988.21916

[66] Farzad Parvaresh and Alexander Vardy. 2005. Correcting Errors Beyond the Guruswami-Sudan Radius in Polynomial Time. In *46th FOCS*. IEEE Computer Society Press, 285–294. https://doi.org/10.1109/SFCS.2005.29

[67] Chris Peikert and Sina Shiehian. 2019. Noninteractive Zero Knowledge for NP from (Plain) Learning with Errors. In *CRYPTO 2019, Part I (LNCS, Vol. 11692)*, Alexandra Boldyreva and Daniele Micciancio (Eds.). Springer, Heidelberg, 89–114. https://doi.org/10.1007/978-3-030-26948-7_4

[68] David Pointcheval and Jacques Stern. 1996. Provably Secure Blind Signature Schemes. In *ASIACRYPT'96 (LNCS, Vol. 1163)*, Kwangjo Kim and Tsutomu Matsumoto (Eds.). Springer, Heidelberg, 252–265. https://doi.org/10.1007/BFb0034852

[69] Oded Regev. 2003. New lattice based cryptographic constructions. In *35th ACM STOC*. ACM Press, 407–416. https://doi.org/10.1145/780542.780603

[70] Atri Rudra and Mary Wootters. 2018. Average-radius list-recoverability of random linear codes. In *29th SODA*, Artur Czumaj (Ed.). ACM-SIAM, 644–662. https://doi.org/10.1137/1.9781611975031.42

[71] Michael Sipser and Daniel A. Spielman. 1994. Expander Codes. In *35th FOCS*. IEEE Computer Society Press, 566–576. https://doi.org/10.1109/SFCS.1994.365734

[72] Salil P. Vadhan. 2012. *Pseudorandomness*. Now Publishers Inc. https://people.seas.harvard.edu/~salil/pseudorandomness/.