On the Existence of Extractable One-Way Functions

Citation: Bitansky, Nir et al. "On the Existence of Extractable One-Way Functions." SIAM Journal on Computing 45.5 (2016): 1910–1952. © 2016 by SIAM

As Published: http://dx.doi.org/10.1137/140975048

Publisher: Society for Industrial and Applied Mathematics

Persistent URL: http://hdl.handle.net/1721.1/107895

Version: Final published version: final published article, as it appeared in a journal, conference proceedings, or other formally published context

SIAM J. COMPUT.
Vol. 45, No. 5, pp. 1910–1952

ON THE EXISTENCE OF EXTRACTABLE ONE-WAY FUNCTIONS[*]

NIR BITANSKY[†], RAN CANETTI[‡], OMER PANETH[‡], AND ALON ROSEN[§]

Abstract. A function f is *extractable* if it is possible to algorithmically "extract," from any adversarial program that outputs a value y in the image of f, a preimage of y. When combined with hardness properties such as one-wayness or collision-resistance, extractability has proven to be a powerful tool. However, so far, extractability has not been explicitly shown. Instead, it has only been considered as a nonstandard *knowledge assumption* on certain functions. We make headway in the study of the existence of extractable one-way functions (EOWFs) along two directions. On the negative side, we show that if there exist indistinguishability obfuscators for circuits, then there do not exist EOWFs where extraction works for any adversarial program with auxiliary input of unbounded polynomial length. On the positive side, for adversarial programs with bounded auxiliary input (and unbounded polynomial running time), we give the first construction of EOWFs with an explicit extraction procedure, based on relatively standard assumptions (such as subexponential hardness of learning with errors). We then use these functions to construct the first 2-message zero-knowledge arguments and 3-message zero-knowledge arguments of knowledge, against verifiers in the same class of adversarial programs, from essentially the same assumptions.

Key words. knowledge, extraction, zero knowledge, obfuscation

AMS subject classification. 68Q01

DOI. 10.1137/140975048

1. Introduction.

The ability to argue about what adversarial programs "know" in the context of a given interaction is central to modern cryptography. A primary facet of such argumentation is the ability to *efficiently extract* knowledge from the adversarial program. Establishing this ability is often a crucial step in security analysis of cryptographic protocols and schemes.

Cryptographic proofs of knowledge for NP languages are an obvious example for the use of knowledge extraction. In such proof systems, "knowledge of a witness" is *defined* by way of existence of an efficient extraction procedure that given a convincing prover for an NP statement can find a corresponding witness. The ability to extract values from the adversary is also useful for asserting secrecy properties by simulating the adversary's view of an execution of a given protocol, as in the case of zero-knowledge protocols [GMR89] or multiparty computation [GMR89, GMW87]. Other contexts are mentioned within.

[*] Received by the editors June 30, 2014; accepted for publication (in revised form) August 16, 2016; published electronically October 27, 2016. This is a merged version of two works: "How to construct extractable one-way functions against uniform adversaries" (IACR Eprint 2013/468) and "Indistinguishability obfuscation vs. Extractable one-way functions: One must fall" (IACR Eprint 2013/641). An extended abstract of this work appears in *Proceedings of STOC* 2014.

http://www.siam.org/journals/sicomp/45-5/97504.html

Funding: The first author was supported by an IBM Ph.D. fellowship and the Check Point Institute for Information Security; the research was done while the author was at Tel Aviv University. The second author was supported by the Check Point Institute for Information Security, an NSF EAGER grant, and NSF Algorithmic Foundations grant 1218461. The third author was supported by the Simons award for graduate students in theoretical computer science and NSF Algorithmic Foundations grant 1218461. The fourth author was supported by ISF grant 1255/12 and by the ERC under the EU's Seventh Framework Programme (FP/2007-2013), ERC grant agreement 307952.

[†] MIT, Cambridge, MA 02142 (nirbitan@csail.mit.edu).

[‡] Boston University, Boston, MA 02215 (canetti@bu.edu, omer@bu.edu).

[§] Efi Arazi School of Computer Science, IDC Herzliya, Israel (alon.rosen@idc.ac.il).

How is knowledge extracted. Traditionally, the basic technique for extracting knowledge from an adversary is to run it on multiple related inputs to deduce what it "knows" from the resulting outputs. The power of this technique (often called "rewinding") is in that it treats the adversary as a black-box without knowing anything regarding its "internals." However, as a number of impossibility results for black-box reductions and simulation show, this technique is also limited. One main limitation of rewinding-based extraction is that it requires multiple rounds of interaction with the adversary. For example, proving security of candidate 3-message zero-knowledge protocols, succinct noninteractive arguments for NP languages (SNARGs),[1] and other tasks are out of the technique's reach [GK96, GW11].

Starting with the work of Barak [Bar01], a handful of extraction techniques that go beyond the limitations of black-box extraction have been developed. These techniques use the actual adversarial program in an essential way, rather than only the adversary's input-output functionality. However, these too require several rounds of protocol interaction, so they do not work in the above contexts.

Knowledge assumptions and extractable functions. Damgård [Dam92] proposes an alternative approach to knowledge extraction in the form of the *knowledge of exponent assumption (KEA)*. The assumption essentially states that it is possible to extract the secret value x from any program that, given two random generators g, h of an appropriate group G, outputs a pair of group elements of the form g^x, h^x. This approach was then abstracted by Canetti and Dakdouk [CD08, CD09], who formulated a notion of *extractable functions*. These are function families $\{f_e\}$ where, in addition to standard hardness properties, such as one-wayness or collision-resistance, any (possibly adversarial) program $\mathcal{A}$ that given e outputs y in the image of f_e has an "extractor" $\mathcal{E}$ (depending on the code of $\mathcal{A}$) that given e, and any randomness used by $\mathcal{A}$, outputs a preimage of y.

Extractable functions provide an alternative (albeit nonexplicit) "extraction method" that does not rely on interaction with the adversary. As an expression of the method's power, KEA [HT98, BP04a], or even general extractable one-way functions (EWOFs) [CD09, BCC$^+$13], are known to suffice for constructing 3-message zero-knowledge (ZK) protocols, and extractable collision-resistant hash functions (ECRHs) are known to suffice for constructing succinct noninteractive arguments [BCCT12]. KEA had also led to relatively efficient constructions of encryption schemes with strong security guarantees (such as resilience to *chosen cipertext attacks*) [Dam92, BP04a].

The black-box impossibility of some of the above applications implies that it is impossible to obtain extractable functions where the extractor uses the adversary's program $\mathcal{A}$ only as a black-box. Coming up with suitable non-black-box techniques has been the main obstacle in constructing extractable functions, and prior to this work, no construction with an explicit extraction procedure was known. Instead, for all existing candidate constructions of extractable functions (e.g., [Dam92, CD09, BCCT12, BC12]), the existence of such an extractor is merely *assumed.* Such assumptions are arguably not satisfying. For one, they do not qualify as "efficiently falsifiable" [Nao03]; that is, unlike standard assumptions, here it may not be possible to algorithmically test whether a given adversary breaks the assumption. In

[1] Roughly speaking, these are 2-message computationally sound proof systems, where verification can be done much faster than traditional NP verification. Such proof systems have drawn a lot of attention in recent years due to the powerful solution they suggest to the problem of verifiably delegating computation.

addition, the impossibility of extractable functions with black-box extraction only further decreases our confidence in such assumptions, as our current understanding of non-black-box techniques and their limitations is quite limited.

Thus, a natural question arises:

> *Can we construct extractable functions from standard hardness assumptions?*
>
> *Alternatively, can we show that extractable functions cannot exist?*

On the role of auxiliary input. It turns out that the question is more nuanced. Specifically, we show that the answer crucially depends on how we model the "auxiliary information" available to the evaluator $\mathcal{A}$ and the extractor $\mathcal{E}$. Let us elaborate. One straightforward formulation of extractable functions requires that for any possible adversary (modeled as a uniform polynomial-time algorithm) there exists an extractor (again, modeled as a uniform polynomial-time algorithm) that successfully extracts as described above given the adversary's coin tosses. An alternative is to model both the adversary and the extractor as nonuniform families of deterministic polynomial-size circuits.

However, it turns out that in many applications neither formulation suffices. Indeed, when using extractable functions with other components in a larger cryptographic scheme or protocol, an adversary $\mathcal{A}$ may gather information z from other components and use it as *additional* auxiliary input when evaluating the extractable function. To be useful in these cases, the extractor needs to be able to deal with auxiliary information that is determined *after* the extractor has been fixed. That is, we require that for any adversary $\mathcal{A}$ there exists an extractor $\mathcal{E}$ such that for any polynomial-size auxiliary input z, and for a randomly chosen key e, whenever $\mathcal{A}(z, e)$ outputs an image y, $\mathcal{E}(z, e)$ output a corresponding preimage of y.[2] In the above, we can model both the adversary $\mathcal{A}$ and the extractor $\mathcal{E}$ either as a uniform polynomial-time algorithm or as nonuniform algorithms with polynomial-size advice. We call z the *common auxiliary input*, and if $\mathcal{A}$ and $\mathcal{E}$ are nonuniform we refer to their advice as *individual auxiliary input*.

We note that the concept of common auxiliary input appears elsewhere in cryptography. For instance, to make sure that ZK protocols remain ZK under sequential composition, the verifier and simulator get common auxiliary input [Gol93, GO94]. To obtain this standard formulation of ZK using extractable functions, extractability with common auxiliary input is needed. In other settings, the definition can be relaxed to consider only the case where the common auxiliary input is taken from some specific distribution that captures the "possible" auxiliary information in a given system. For example, Goldreich [Gol93] considers the case that the auxiliary input is generated by a uniform polynomial-time algorithm. Another more specific example is in [BCCT12], where they consider a distribution of auxiliary inputs consisting of "randomized encryptions of random messages."

1.1. Overview of results. We give two quite different answers to the above question. On the negative side, following the common belief (first expressed in [HT98]), we give formal evidence that EWOFs with common auxiliary input of *unbounded* length may not exist. Specifically, we show that such extractable functions cannot exist assuming a notion of program obfuscation called *indistinguishability obfuscation* (IO) [BGI+01].[3]

[2] Above we avoid explicitly referring to the adversary's coin tosses. Any such coins will be considered as part of the auxiliary input z.

[3] We further elaborate on the notion of IO later in this introduction.

Theorem 1 (informal). *If there exist indistinguishability obfuscators for circuits, then there do not exist EWOFs with respect to common auxiliary input of unbounded polynomial length.*

This seems to suggest that the concept of EWOFs (and other concepts that imply it, such as extractable collision-resistant hashing or SNARGs) may be shaky overall. Indeed, the notion of IO is not known to be subject to any lower bounds or limitations, and starting from the work of [GGH$^+$13b], several candidate constructions of IO have been exhibited (so far, based on strong assumptions).

On the positive side, we show, for the first time, how to construct extractable one-way functions with an explicit extraction procedure with respect to auxiliary input of *bounded* polynomial length (common or individual) and, in particular, with respect to *uniform adversaries*. More specifically, we first give a construction of extractable one-way functions based on verifiable delegation schemes for deterministic computations, which we will call *publicly verifiable delegation schemes*.[4]

Theorem 2 (informal). *Assuming one-way functions and publicly verifiable delegation schemes, there exist EOWFs with respect to auxiliary input of bounded polynomial length (common or individual).*

The existence of publicly verifiable delegation schemes is perhaps not considered a standard assumption, but it is a falsifiable assumption (in the sense of [Nao03], which we described above).[5] Furthermore it has candidates such as Micali's *CS proofs* [Mic00], SNARGs [BCCT13], or constructions based on IO [BGL$^+$15, CHJV15, KLW15]. We view this construction mainly as a proof of concept, showing that ruling out such extractable functions may be a hard task.

Aiming toward a construction from standard assumptions, we formulate a generalized variant of EWOFs (GEOWFs), capturing the properties which make EOWFs useful, and indeed construct bounded-auxiliary-input GEOWFs from relatively standard assumptions. Specifically, our construction relies on the existence of *privately verifiable* delegation schemes, first established by [KRR14], based, for instance, on the learning with errors assumption. We additionally show that the limitation given by Theorem 1 also holds for GEOWFs.

Relying on GEOWFs, we give the first constructions from standard assumptions of 2-message ZK arguments and 3-message ZK arguments of knowledge, against verifiers with bounded auxiliary input.[6]

Theorem 3 (informal).

1. *Assuming (even privately verifiable) delegation schemes, there exist GEOWFs with respect to auxiliary input of bounded polynomial length (common or individual).*
2. *Assuming GEOWFs, ZAPs* [DN07]*, and (*1*-hop* [GHV10]*) homomorphic encryption, there exists a* 3*-message ZK argument of knowledge against bounded-auxiliary-input verifiers. Assuming the GEOWFs are one-way against subexponential-time adversaries, there exists a* 2*-message zero-knowledge argument against bounded-auxiliary-input verifiers.*

We next elaborate on each of the results.

[4] Below we describe this notion more elaborately.

[5] See the discussion in [CLP13] on the equivalent concept of 2-message P-certificates.

[6] By *arguments* we mean computationally sound [BCC88].

1.2. Impossibility with respect to unbounded auxiliary input. To introduce the negative result regarding EOWFs with unbounded (common) auxiliary input, we first elaborate on the concept of obfuscation and explain its contrast with auxiliary-input extractability.

Obfuscation. Program obfuscation is aimed at making code unintelligible while preserving its functionality and has been long considered the holy grail of cryptography, with diverse and far-reaching applications. Barak et al. [BGI$^+$01] initiated the rigorous treatment of obfuscation, formulating a number of definitions of security for the task. However, until recently, we only knew how to obfuscate a number of restricted classes of programs under *any* of these definitions. Furthermore, Barak et al. demonstrated a class of programs that are *unobfuscatable* according the natural virtual black-box (VBB) definition, guaranteeing that access to the obfuscated program gives no more power than access to an impenetrable black box with the same input-output functionality.

This state of affairs changed with the work by Garg et al. [GGH$^+$13b], who proposed a candidate construction of general-purpose obfuscators. They show that under strong assumptions on multilinear maps [GGH13a], their construction satisfies the relaxed notion of IO [BGI$^+$01], for which no impossibility results are known. The IO notion only requires that it is hard to distinguish an obfuscation of C_0 from an obfuscation of C_1 for any two circuits C_0 and C_1 of the same size that compute the exact same function.

The tension between obfuscation and extractable functions. As noted already in the work of Hada and Tanaka [HT98], extractability with respect to common auxiliary input is a strong requirement. Indeed, the common auxiliary input z may potentially encode an arbitrary circuit to be executed by the adversary in order to produce an image y. The extractor should, thus, be able to efficiently "reverse engineer" such a circuit, in order to figure out a preimage of y. This reveals a clear tension with obfuscation: if z contains obfuscated code that chooses a preimage in some complicated way, it may be impossible to extract from.

The question is how to turn this intuition into a formal impossibility. While VBB obfuscation may be the natural choice, we do not have any evidence that there exist VBB obfuscators for a complicated task such as the one described above (in fact, there is evidence that they do not [GK05, BCC$^+$14]). We show that IO suffices to make this intuition rigorous.

Proof idea. We focus on the "hardest scenario," where the auxiliary input z may represent an arbitrary malicious and potentially obfuscated code. Specifically, we consider the following folklore case (sketched in [BCCT12]), where z is an obfuscation of a circuit C_k that, given key e for an extractable function f_e, chooses its preimage in an unpredictable way: it applies PRF_k, a *pseudorandom function* [GGM86], to the key and outputs the result $f_e(\mathsf{PRF}_k(e))$.

An adversary, given such an obfuscated circuit as auxiliary input z, can run it on the key e for the extractable function and always obtain a proper image. The question is whether the extractor, given the same (e, z), can output a preimage. Intuitively, had we given the extractor black box access to the circuit C_k, instead of an obfuscation of C_k, it would have to invert the one-way function to obtain such a preimage. Indeed, since the oracle C_k answers any query e' with $f_{e'}(\mathsf{PRF}_k(e'))$, it follows from pseudorandomness that finding a preimage of $f_e(\mathsf{PRF}_k(e))$ is as hard as finding a preimage of $f_e(u)$ for a uniformly random u.

Can the above intuition be translated to a proof using IO? Indeed, when z is an IO obfuscation $i\mathcal{O}(C_k)$ of the circuit C_k, it is not clear what kind of information leaks on the PRF key k. Nevertheless, we show that the above intuition can still be fulfilled. The idea is to consider an alternative to the circuit C_k that computes the same function, but without actually "knowing" the preimage $\mathsf{PRF}_k(e)$. This is achieved using the *puncturing technique* of Sahai and Waters [SW14].

Specifically, instead of using any PRF family, we use a *puncturable PRF*. In such PRFs it is possible to puncture a given key k at an arbitrary point x^* in the domain of the function. The punctured function $\mathsf{PRF}_{k_{x^*}}$, with punctured key k_{x^*}, preserves functionality at any other point, but hides any information on the point $\mathsf{PRF}_k(x^*)$; namely, the value $\mathsf{PRF}_k(x^*)$ is pseudorandom, even given (x^*, k_{x^*}). As shown in several works [BW13, BGI14, KPTZ13], such puncturable PRFs follow from the GGM construction [GGM86].

Using a puncturable PRF in the implementation of C_k, we can now show that if the extractor succeeds in finding a preimage of $y = f_e(\mathsf{PRF}_k(e))$, it would also succeed had we provided it with an obfuscation of an alternative circuit $C_{k_e,y}$. The circuit $C_{k_e,y}$ computes the same function as C_k, but in a different way: it only has the punctured key k_e and has the value $y = f_e(\mathsf{PRF}_k(e))$ directly hardwired into it, so that it does not have to evaluate the PRF in order to compute it. Thus, the fact that the extractor still succeeds follows by the guarantee of IO. However, now by the pseudorandomness guarantee at the punctured point e, we know that $\mathsf{PRF}_k(e)$ is pseudorandom, even given the circuit $C_{k_e,y}$, and thus the extractor can be used to invert the one-way function f_e from scratch.

Finally, we note that since puncturable PRFs can be constructed from one-way functions, and any EOWF is in particular a one-way function, it follows that the impossibility of EOWFs is implied by IO without any further assumptions. We also note that the result naturally extends to the notion of generalized EOWFs (presented in more detail in the following subsection).

So, is the KEA wrong?. In its original formulation [Dam92] and in subsequent formulations [HT98, BP04a, BP04b], the KEA was not stated with respect to common auxiliary input but rather only for individual auxiliary input (or completely uniform algorithms), where any $\mathcal{A}$ with advice $z_{\mathcal{A}}$ has an extractor $\mathcal{E}$ with its own advice $z_{\mathcal{E}}$, and the only common extra information is the adversary's coin tosses and key for the function. In particular, given a nonuniform adversary $\mathcal{A}$ with an obfuscated code as advice $z_{\mathcal{A}}$, the extractor is allowed to have a different advice $z_{\mathcal{E}}$, representing the "deobfuscated" code. Indeed, our result does not rule out such a notion of extraction (even assuming IO for all circuits).

Our result does not invalidate the intuition that "the only way" to compute (g^x, h^x), given (g, h), is by "knowing" x. As we saw, our adversary and auxiliary input are devised so that x is actually known, but only by an underlying obfuscated computation, and thus cannot be figured out efficiently by an external extractor.

We also note that our result does not rule out extractable functions with respect to common auxiliary input that is taken from specific distributions that may be conjectured to be "benign."

Subsequent work. The negative result presented above, in fact, shows that for any candidate EOWF family $\mathcal{F}$, there exists a distribution $\mathcal{Z}_{\mathcal{F}}$, and an adversary $\mathcal{A}$, such that any extractor $\mathcal{E}$ for $\mathcal{A}$ would fail with respect to common auxiliary input sampled from $\mathcal{Z}_{\mathcal{F}}$. As noted by Boyle and Pass [BP15], our result can be generalized so that $\mathcal{Z}$ does not depend on $\mathcal{F}$ but only on some upper bound $T_{\mathcal{F}}$ on its running time (by having $\mathcal{Z}$ encode a proper universal circuit). Boyle and Pass further show that, assuming a strengthening of IO called *extractable obfuscation* (also known as *differing inputs obfuscation*), $\mathcal{Z}$ can be made independent of $T_{\mathcal{F}}$ and only depend on its output length $\ell_{\mathcal{F}}$; in particular, elements sampled from $\mathcal{Z}$ can be longer than $\ell_{\mathcal{F}}$. We note that their result does not clash with our positive result for bounded auxiliary input, in which $\ell_{\mathcal{F}}$ is made longer than the bound on auxiliary inputs. We also note that both our and Boyle and Pass's impossibility apply for a specific and rather contrived distribution. No impossibility is yet known for distributions that may be considered "benign," such as the uniform distribution.

1.3. Constructions with respect to bounded auxiliary input. We first formulate GEOWFs and show how GEOWFs can be constructed from standard assumptions. Then, we shall see that under appropriate conditions, we can leverage the same ideas in order to get standard EOWFs.

Generalized EOWFs. The essence of EOWFs, and what makes them useful, is the asymmetry between a black-box inverter and a non-black-box extractor: an inverter, which only gets a random image $y = f_e(x)$ of an EOWF, cannot find a corresponding preimage x', whereas a non-black-box extractor, which is given a code that produces such an image, can find a preimage x'. GEOWFs allow us to express this asymmetry in a more flexible way. Concretely, a function family $\mathcal{F}$ is now associated with a "hard" binary relation $\mathcal{R}_e^{\mathcal{F}}$ on image-witness pairs $(f_e(x), x')$. Given $y = f_e(x)$ for a random x, it is infeasible to find a witness x' such that $\mathcal{R}_e^{\mathcal{F}}(y, x') = 1$. In contrast, a non-black-box extractor that is given a code that produces such an image can find such a witness x'.

It is natural to require that the relation $\mathcal{R}_e^{\mathcal{F}}$ is efficiently testable; in this case we say that the GEOWF is *publicly verifiable.* However, we shall see that GEOWFs are useful, even for hard relations that are not publicly verifiable. Specifically, we will consider *privately verifiable* GEOWFs where $\mathcal{R}_e^{\mathcal{F}}(y, x')$ is not efficiently testable given only $(y = f_e(x), x')$ but can be efficiently tested given x in addition.

The main idea behind the construction. To convey the basic idea behind our constructions of GEOWFs with respect to bounded auxiliary input, consider the following first attempt. The GEOWF f is keyless; it is simply a pseudorandom generator (PRG) stretching inputs of length n to outputs of length $2n$. The relation $\mathcal{R}^{\mathcal{F}}$ contains pairs $(y, \mathcal{M})$ such that the witness $\mathcal{M}$ is a description of a machine of length at most n and $\mathcal{M}(1^n)$ outputs y. The fact that the relation $\mathcal{R}^{\mathcal{F}}(y, \cdot)$ is hard to satisfy for $y = f(x)$ and a random x follows from the pseudorandomness of the output y. Indeed, a truly random output that is indistinguishable from y would have high Kolmogorov complexity. However, given any adversarial program $\mathcal{M}_{\mathcal{A}}$ whose description size is bounded by n and that outputs some $y \in \{0,1\}^{2n}$, the description of the program $\mathcal{M}_{\mathcal{A}}$ itself is a witness that satisfies the relation $\mathcal{R}^{\mathcal{F}}(y, \mathcal{M}_{\mathcal{A}})$, and thus extraction is trivial.

The main problem with the above is that the time required to test the relation $\mathcal{R}^{\mathcal{F}}$ (even given some preimage of y) is not bounded by any particular polynomial; indeed, the running time of $\mathcal{M}_{\mathcal{A}}$ may be an arbitrary polynomial. One can try to fix this by padding the witness $\mathcal{M}_{\mathcal{A}}$ with 1^t, where t is the running time of $\mathcal{M}_{\mathcal{A}}$. However, now the length of the extracted witness depends on the running time of the adversarial program $\mathcal{M}_{\mathcal{A}}$ and is not bounded by any particular polynomial in the length of the image. Such generalized extractable functions do not seem to be as powerful, though; in particular, we do not know how to use them in applications such as 2-message and 3-message ZK protocols.

A similar problem is encountered in Barak's ZK protocol [Bar01], where the entire computation of a malicious verifier is used as a "trapdoor" in the ZK simulation procedure. Inspired by the ZK protocol of Barak, Lindell, and Vadhan [BLV06], we get around this problem using a non-interactive proof system that allows for *quick* *verification* of (possibly long) computations. Instead of computing the output y of the witness program $\mathcal{M}_\mathcal{A}$, $\mathcal{R}^\mathcal{F}$ will (quickly) verify a proof for the fact that $\mathcal{M}_\mathcal{A}(1^n)$ outputs y. That is, $(y, (\mathcal{M}, \pi)) \in \mathcal{R}^\mathcal{F}$ only if π is a convincing proof that $\mathcal{M}(1^n) = y$. Intuitively, the soundness of the proof guarantees that the relation is still hard to satisfy. Extraction from a bounded-auxiliary-input adversary $\mathcal{M}_\mathcal{A}$ is done by computing a proof for its computation.

Delegation schemes. The proof system required in our constructions is a noninteractive computationally sound proof for deterministic statements, from here on referred to simply as a delegation scheme. More precisely, in a delegation scheme, the verifier generates, once and for all, an "offline message" σ together with a private verification state τ, and sends σ to the prover. Then, the prover can compute a noninteractive proof π for any adaptively chosen statement of the sort: "machine $\mathcal{M}$ outputs v within t steps." We require that the verifier runs in time polynomial in the security parameter n, but only polylogarithmic in t, and the prover runs in time polynomial in (t, n). We say that a delegation scheme is *publicly verifiable* if the verification state τ can be published without compromising soundness. Otherwise we say that the scheme is *privately verifiable*.

As mentioned in section 1.1, while we do have candidates for publicly verifiable delegation schemes, their security is not based on standard assumptions. In a recent breakthrough result, Kalai, Raz, and Rothblum [KRR14] construct a privately verifiable delegation scheme based on any private information retrieval scheme with subexponential security. The scheme of [KRR14] has two deficiencies that we need to deal with. First, it only has nonadaptive soundness, and second they require that a bound on the running time t of proven computations is provided at setup time when (τ, σ) are generated. To overcome the first problem, we show that relying on complexity leveraging soundness can be enhanced for statements that are adaptively chosen from a relatively small set of possible statements, which will suffice for our purposes. We show that the second problem can be easily solved as long as t is exponentially bounded in the security parameter. In the body of the paper, we call such delegation schemes *universal* (in the spirit of [BG08]) to capture the fact that a single system accounts for computations of a priori unbounded polynomial time.

GEOWFs from delegation schemes. We now sketch how delegation schemes are used in our constructions. We obtain publicly verifiable (respectively, privately verifiable) GEOWFs based on publicly verifiable (respectively, privately verifiable) delegation schemes. In both cases, the GEOWF f is keyless, it is given as input a seed s and a random string r. f applies a PRG on s and obtains an image v. f then uses the randomness r to sample an offline message σ together with a verification state τ for a delegation scheme. Finally, f outputs (v, σ). We assume that if the delegation scheme is publicly verifiable, the offline message σ includes the verification state τ. Also, if the delegation scheme is privately verifiable, we assume that τ can be (perhaps inefficiently) determined from σ. (Both assumptions are without loss of generality (w.l.o.g).)

The relation $\mathcal{R}^\mathcal{F}$ contains pairs consisting of an image (v, σ) and witness $(\mathcal{M}, \pi)$, such that the length of $\mathcal{M}$ is much shorter than the length of v, and π is an accepting proof for the statement "$\mathcal{M}(1^n)$ outputs v," with respect to the verification state τ corresponding to the offline message σ. Indeed, if the delegation scheme is publicly verifiable, τ can be efficiently computed from σ, and therefore the relation $\mathcal{R}^\mathcal{F}$ is efficiently testable. And if the delegation scheme is privately verifiable, τ can be efficiently computed given a primage of (v, σ) that contains the randomness used to sample σ and τ.

Constructing standard EOWFs. We show how to construct a standard (not generalized) EOWF g from a publicly verifiable GEOWF f. The basic high-level idea is to embed the structure of the GEOWF f and the relation $\mathcal{R}^{\mathcal{F}}$ into the standard EOWF g. For this purpose, g will get as input a string $i \in \{0,1\}^n$, which intuitively picks one of two branches for computing the function. If $i \neq 0^n$, which is almost always the case for a random input, the output is computed in the "normal branch," where g takes an input x for the GEOWF f and outputs $f(x)$. If $i = 0^n$, the output is computed in the "trapdoor branch," which is almost never taken for a random input but is used by the extractor. In the trapdoor branch, g takes as input a candidate output y for f and a witness x' for $\mathcal{R}^{\mathcal{F}}(y,\cdot)$. g verifies that $(y,x') \in \mathcal{R}^{\mathcal{F}}$ and if so, it outputs y. Given an adversarial program $\mathcal{M}_{\mathcal{A}}$ that outputs y in the image of f, the extractor for g can invoke the extractor for f, obtain a witness x' such that $(y,x') \in \mathcal{R}^{\mathcal{F}}$, and from this witness construct a valid (trapdoor branch) primage $(i = 0^n, y, x')$ for y.

The above transformation cannot start from a privately verifiable GEOWF; indeed public verification is required to allow the function to efficiently evaluate the relation $\mathcal{R}^{\mathcal{F}}$ in the trapdoor branch. We also note that the above transformation is oversimplified and implicitly assumes that an adversarial evaluator cannot use the trapdoor branch of the function to produce an output that is in the image of g but not in the image of f, in which case extraction may fail. In the paper we show how to avoid this problem by relying on the specific construction of publicly verifiable GEOWFs from publicly verifiable delegation schemes that possess an extra property (satisfied by existing candidates).

1.4. Zero knowledge against verifiers with bounded auxiliary input.

We start by describing how to construct 2-message and 3-message ZK protocols from standard (nongeneralized) EOWFs, and then explain how to replace the EOWFs with GEOWFs.

From EOWFs to 3*-message ZK.* The protocol follows a common methodology for designing ZK protocols known as *the Feige–Lapidot–Shamir trapdoor paradigm* [FLS99]. Given, say a keyless, EOWF f, the basic idea is to have the verifier send the prover an image $y = f(x)$ of a random element x, which will serve as the trapdoor. The prover would then give a witness-indistinguishable (WI) proof of knowledge (WIPOK) attesting that it either knows a witness w for the proven statement or knows a preimage x' of y. Intuitively, soundness (and actually proof of knowledge) follow from the one-wayness of f and the proof of knowledge property of the WI system. Zero knowledge follows from the extractability of f. Indeed, the simulator, given the code of the verifier, can run the extractor of the EOWF, obtain x, and use it in the WI proof.

Following through on this intuition encounters several difficulties. First, a WIPOK requires three messages, and thus a first WI prover message must be sent in the first message of the protocol. Furthermore, the WI statement is determined only when the verifier sends y in the second protocol message. Therefore, we must make sure to use a WIPOK where the first prover message does not depend on the statement. Another basic problem concerns the length of the first WI message. Recall that, in our construction of EOWFs against bounded auxiliary input adversaries, the function's output is longer than the adversary's advice. Since a cheating verifier may compute y using the first WI message as advice, we must use a WI system where the length of the first message is independent of the length of the proven statement. We design a WI argument with the required properties based on two standard cryptographic primitives: ZAPs [DN07] and extractable commitments [PW09].

An additional potential problem is that a malicious verifier may output an element $\tilde{y}$ outside of the function's image, an event which in general may not be efficiently recognizable, and cause the simulator to fail. This can be solved in a couple of generic ways; below we outline one such solution, based on 1-hop homomorphic encryption. A different approach to the problem, based on ZAPs, is described in [BCC$^+$13].

From EOWFs to 2*-message ZK.* In the 2-message protocol, we replace the 3-message WIPOK with a 2-message WI proof (e.g., a ZAP). However, in the above 3-message protocol, soundness is established by using the POK property of the WI, whereas 2-message WI proofs of knowledge are not known. Instead, relying on ideas similar to those used in [BLV06], we prove soundness using complexity leveraging. The prover adds to its message a statistically binding commitment to an arbitrary message and proves that either "$x \in \mathcal{L}$" or "$f(x) = y$ *and the commitment opens to* x." We require that the commitment is invertible in some superpolynomial time T, whereas the one-wayness of f still holds against adversaries that run in time poly(T). Now, an inverter of f can run the cheating prover with a verifier message that contains its input image y and brute-force break the commitment to obtain a preimage of y.

Replacing EOWF with GEOWF. We would like to base our ZK protocols on privately verifiable GEOWFs (which can be constructed from standard assumptions) instead of on EOWFs. A natural first attempt is to modify the protocol as follows: the verifier sends an image $y = f(x)$, as before, and the prover then gives a WIPOK attesting either that it knows a witness w for the proven statement or that it knows not a preimage but a witness x' such that $\mathcal{R}^{\mathcal{F}}(y, x') = 1$. The main problem with this first attempt is that the relation $\mathcal{R}^{\mathcal{F}}$ is not publicly verifiable, and thus the simulator has no way of proving the statement. Another possible problem is that a malicious verifier may output an element outside of the function's image, an event which in general may not be efficiently recognizable. In such a case there is no extraction guarantee, and simulation may fail.

The solution for both problems is to test the relation $\mathcal{R}^{\mathcal{F}}$, and the validity of the verifier's image, using a 2-message secure function evaluation protocol, based, for example, on a 1-hop homomorphic encryption [GHV10]. More concretely, the verifier, in addition to the function's output y, sends an encryption c of the input x. The simulator then homomorphically evaluates a circuit that efficiently computes $\mathcal{R}^{\mathcal{F}}(y, x')$ given x, as well as verifies that indeed $y = f(x)$. The simulator then obtains an evaluated ciphertext $\hat{\mathsf{c}}$ that decrypts to 1 (the honest prover will simply simulate an encryption $\hat{\mathsf{c}}$ of 1). Finally, the prover (or simulator) sends back $\hat{\mathsf{c}}$ and gives a WIPOK attesting either that it knows a witness w for the proven statement or that the ciphertext $\hat{\mathsf{c}}$ was generated as described. The verifier verifies the WI proof is accepting and that $\hat{\mathsf{c}}$ decrypts to 1.

Limitations on 2*- and* 3*-message ZK and related work.* Three-message ZK protocols with black-box simulation exist only for trivial languages [GK96]. The impossibility extends to the case of adversaries with bounded advice of size $n^{\Omega(1)}$, where n is the security parameter (see Appendix A for more details). Previous 3-message ZK protocols were based either on the KEA [HT98, BP04a], on EOWF [BCC$^+$13], or on other extractability assumptions [CD08]. In all, the simulator uses a nonblack extractor that is only assumed to exist but not explicitly constructed.

2-message ZK arguments against adversaries with unbounded polynomial advice exist only for trivial languages (regardless of how simulation is done) [GO94]. In fact, this impossibility extends even to adversaries with bounded advice, provided that the advice string is longer than the verifier's message. Barak, Lindell, and Vadhan

[BLV06] construct a 2-message argument that is ZK as long as the verifier's advice is shorter than the verifier message by a superlogarithmic additive factor. Indeed, our 2-message protocol has the same skeleton. However, security of the Barak–Lindell–Vadhan protocol is only shown assuming existence of delegation schemes (or, in fact, universal arguments for nondeterministic languages) that are *publicly verifiable*, which as discussed earlier is not considered to be a standard assumption.

1.5. Open questions. This work leaves open several questions regarding the existence of extractable function. We next highlight some of these questions that we find mostly intriguing:

1. There is a gap between the positive and negative results in terms of the type and length of auxiliary input. Specifically, we do not know if there exist EOWFs with respect to individual auxiliary input of unbounded polynomial length and no common auxiliary input (or common auxiliary input of bounded polynomial length).
2. Another question regards the existence of extractable functions (even with respect to completely uniform adversaries) that satisfy stronger one-wayness properties. Particularly interesting is the possibility of extractable functions where the adversary's output computationally binds it to a specific input, for example, ECRHs and extractable injective one-way functions.
3. Finally, we ask whether there exist EOWFs with respect to common auxiliary input that is taken from specific "benign" distribution, such as the uniform distribution.

Organization. In section 2, we recall basic conventions and notation. In section 3, we give the relevant definitions for EOWF and GEOWF. In section 4, we present the limitation on unbounded auxiliary-input EOWFs based on IO. In section 5, we present the constructions of bounded-auxiliary-input EOWFs and GEOWFs. In section 6.4, we present the ZK protocols constructed from GEOWFs. In Appendix A, we discuss relevant *black-box lower bounds* for EOWFs and ZK.

2. Preliminaries.

Standard computational concepts and conventions. We recall the standard notions that we rely on.

- We say that a (uniform) Turing machine is PPT if it is probabilistic and runs in polynomial time. A polynomial-size circuit family $\mathcal{C}$ is a sequence of circuits $\mathcal{C} = \{C_n\}_{n\in\mathbb{N}}$ such that each circuit C_n is of polynomial size $n^{O(1)}$ and has $n^{O(1)}$ inputs and outputs bits.
- We follow the standard habit of modeling any efficient adversary as a family of polynomial-size circuits. For an adversary $\mathcal{A}$ corresponding to a family of polynomial-size circuits $\{\mathcal{A}_n\}_{n\in\mathbb{N}}$ we often omit the subscript n when it is clear from the context.
- We say that a function $\nu(\cdot)$ is negligible if it decays faster than any inverse polynomial in n. We may also denote this by $\nu(n) = n^{-\omega(1)}$. Following common practice, we denote by $\mathrm{negl}(n)$ any nonspecific negligible function (typically these functions will describe the success probability of an adversary in a specific security definition and are allowed to depend on the adversary in question).
- For an infinite index set I, we say that two distribution ensembles $\mathcal{X} = \{\mathcal{X}_i\}_{i\in I}$ and $\mathcal{Y} = \{\mathcal{Y}_i\}_{i\in I}$ are computationally indistinguishable and denote

this by $\mathcal{X} \approx_c \mathcal{Y}$ if for any polynomial-size distinguisher $\mathcal{D} = \{\mathcal{D}_n\}_{n\in\mathbb{N}}$, any $n \in \mathbb{N}$ and $i \in I \cap \{0,1\}^n$,

$$|\Pr[\mathcal{D}_n(X_i) = 1] - \Pr[\mathcal{D}_n(\mathcal{Y}_i) = 1]| \leq \text{negl}(n) \enspace .$$

- We denote by U_n the uniform distribution on strings in $\{0,1\}^n$.

Proof systems and arguments. We consider (possibly interactive) proof systems between a prover P and a PPT verifier V. We denote the output of the verifier by $\langle P(a), V(b)\rangle(c)$, where the parties have individual inputs a and b (respectively) and common input c.

For a relation $\mathcal{R}_\mathcal{L}$ corresponding to a language $\mathcal{L} = \{\varphi \mid \exists w \text{ such that } (\varphi, w) \in \mathcal{R}_\mathcal{L}\}$, we say that (P,V) is a proof system for $\mathcal{R}$ (or $\mathcal{L}$) if is satisfies the following:

1. Completeness: For any $\varphi \in \mathcal{L} \cap \{0,1\}^n$,
$$\Pr[\langle P, V\rangle(\varphi) = 1] = 1 \enspace .$$
2. Soundness: For any malicious P^* and $\varphi \in \{0,1\}^n \setminus \mathcal{L}$,
$$\Pr[\langle P^*, V\rangle(\varphi) = 1] \leq \text{negl}(n) \enspace .$$

We say that a system is an argument if soundness is guaranteed only against polynomial-size $P^* = \{P^*_n\}_{n\in\mathbb{N}}$.

3. Extractable one-way functions.

In this section, we define auxiliary-input EOWFs, bounded-auxiliary-input EOWFs, and GEOWFs.

DEFINITION 4 (auxiliary-input EOWFs [CD08]). *Let ℓ, ℓ', m be polynomially bounded length functions. An efficiently computable family of functions*

$$\mathcal{F} = \left\{ f_e : \{0,1\}^{\ell(n)} \to \{0,1\}^{\ell'(n)} \;\middle|\; e \in \{0,1\}^{m(n)}, n \in \mathbb{N} \right\} \enspace ,$$

associated with an efficient (probabilistic) key sampler $\mathcal{K}_\mathcal{F}$, is an auxiliary-input EOWF if it is as follows:

1. One-way: *For any polynomial-size $\mathcal{A} = \{\mathcal{A}_n\}_{n\in\mathbb{N}}$ and $n \in \mathbb{N}$,*
$$\Pr_{\substack{e\leftarrow\mathcal{K}_\mathcal{F}(1^n)\\ x\leftarrow\{0,1\}^{\ell(n)}}}\left[\begin{array}{c} x' \leftarrow \mathcal{A}(e, f_e(x)) \\ f_e(x') = f_e(x) \end{array}\right] \leq \text{negl}(n) \enspace .$$
2. Extractable: *For any PPT adversary $\mathcal{A}$, there exists a PPT extractor $\mathcal{E}$ such that, for any polynomial b, security parameter $n \in \mathbb{N}$, and $z \in \{0,1\}^{b(n)}$,*
$$\Pr_{e\leftarrow\mathcal{K}_\mathcal{F}(1^n)}\left[\begin{array}{c|c} \exists x : f_e(x) = y & y \leftarrow \mathcal{A}(e; z) \\ f_e(x') \neq y & x' \leftarrow \mathcal{E}(e; z) \end{array}\right] \leq \text{negl}(n) \enspace .$$

Bounded auxiliary input. We now define bounded-auxiliary-input EOWFs. Unlike the definition above, where extraction is guaranteed with respect to auxiliary input of any polynomial size b, here b is fixed in advance and the function is designed accordingly. That is, extraction is guaranteed only against adversaries whose advice is bounded by b, whereas their running time may still be an arbitrary polynomial; this, in particular, captures the class of *uniform polynomial-time adversaries.*

For b-bounded auxiliary input, we also define keyless families. While for unbounded auxiliary input, extraction is impossible for keyless families (the adversary may get as auxiliary input a random image, thus forcing the extractor to break one-wayness), for b-bounded auxiliary input, it may be possible, since the output length ℓ' can be larger than the bound b on the auxiliary input. Our constructions will yield such keyless functions.

DEFINITION 5 (b-bounded-auxiliary-input EOWFs). *Let b, ℓ, ℓ', m be polynomially bounded length functions (where ℓ, ℓ', m may depend on b). An efficiently computable family of functions*

$$\mathcal{F} = \left\{ f_e : \{0,1\}^{\ell(n)} \to \{0,1\}^{\ell'(n)} \;\middle|\; e \in \{0,1\}^{m(n)}, n \in \mathbb{N} \right\} \quad ,$$

associated with an efficient (probabilistic) key sampler $\mathcal{K}_{\mathcal{F}}$, is a b-bounded auxiliary-input EOWF if it is as follow:

1. One-way: *As in Definition* 4.
2. Extractable against b-bounded adversaries: *For any PPT adversary $\mathcal{A}$, there exists a PPT extractor $\mathcal{E}$ such that for any security parameter $n \in \mathbb{N}$, and $z \in \{0,1\}^{b(n)}$,*

$$\Pr_{e \leftarrow \mathcal{K}_{\mathcal{F}}(1^n)} \left[\begin{array}{c} \exists x : f_e(x) = y \\ f_e(x') \neq y \end{array} \;\middle|\; \begin{array}{c} y \leftarrow \mathcal{A}(e; z) \\ x' \leftarrow \mathcal{E}(e; z) \end{array} \right] \leq \text{negl}(n) \quad .$$

We say that the function is keyless *if in all the above definitions the key is always set to be the security parameter, namely, $e \equiv 1^n$. In this case, the extraction guarantee always holds (rather than only for a random key).*

Remark 1 (bounded randomness). Throughout, we treat any randomness used by the adversary as part of its advice z; in particular, in the case of bounded advice, we assume that the randomness is bounded accordingly. For many applications, this is sufficient as we can transform any adversary that uses arbitrary polynomial randomness to one that uses bounded randomness, by having it stretch its randomness with a PRG. This approach is applicable, for example, for ZK protocols against b-bounded auxiliary-input verifiers (see section 6), as well as for any application where testing if the adversary breaks the scheme can be done efficiently.

Remark 2 (other forms of auxiliary input).

1. *Individual versus common auxiliary input.* In the above formulation of extractability, the adversary $\mathcal{A}$ (producing an image) and the extractor $\mathcal{E}$ are modeled as uniform PPT machines that obtain the same *common* auxiliary input z. This formulation is aligned with the treatment of auxiliary input in other settings such as ZK or obfuscation and, as explained in the introduction, is instrumental when arguing about extractable functions in the context of a larger system.

   As also mentioned in the introduction, in certain contexts it may be sufficient to consider *individual* auxiliary input, where we only require that for any $\mathcal{A}$ with auxiliary input $z_{\mathcal{A}}$, there exists an extractor $\mathcal{E}$ with auxiliary input $z_{\mathcal{E}}$. The extractor's $z_{\mathcal{E}}$ may arbitrarily and inefficiently depend on $z_{\mathcal{A}}$ and could be of an arbitrary polynomial size. This weaker notion may be useful in cases where the adversary's auxiliary inputs do not depend on computations that may have taken place in the system before the extractable function is used. Examples include constructions of encryption schemes with strong security guarantees (resilience to chosen ciphertext attacks and *plaintext awareness*) based on nonuniform security reductions [Dam92, BP04b]. (We may also consider a definition that allows both individual and common auxiliary input.)
2. *Common but "benign" auxiliary input.* In the above formulation, it is required that extraction works for a worst-case choice of the common auxiliary input z. In certain contexts, however, it is sufficient to consider a definition where the common auxiliary input z is drawn from a specific distribution

that is *conjectured* to be "benign," in the sense that it is unlikely to encode a malicious obfuscation. For instance, the distribution can be uniform or an encryption of a random string. Examples where this is sufficient include many of the works on SNARGs, succinct noninteractive ZK, and targeted malleability that rely on extractable primitives [DCL08, Mie08, Gro10, GLR11, BSW12, BCCT12, BC12, DFH12, Lip12, BCCT13, BCI$^+$13, GGPR13, Lip13].

3.1. Generalized extractable one-way functions. The essence of EOWFs, and what makes them useful is the asymmetry between an inverter and a non-black-box extractor: a black-box inverter that gets only a random image $y = f_e(x)$ cannot find a corresponding preimage x', whereas a non-black-box extractor, which is given a code that produces such an image (including any randomness it uses), can find a preimage x'. GEOWFs allow us to express this asymmetry in a more flexible way. Concretely, a function family $\mathcal{F}$ is now associated with a "hard" relation $\mathcal{R}_e^{\mathcal{F}}(f_e(x), x')$ on image-witness pairs $(f_e(x), x') \in \{0,1\}^{\ell'} \times \{0,1\}^{\ell}$. Given $y = f_e(x)$ for a random x, it is infeasible to find a witness x' such that $\mathcal{R}_e^{\mathcal{F}}(y, x') = 1$. In contrast, a non-black-box extractor that is given a code that produces such an image can find such a witness x'.

We consider two variants of GEOWFs. The first is *publicly verifiable GEOWFs*, where for $(y = f_e(x), x')$ the relation $\mathcal{R}_e^{\mathcal{F}}(y, x')$ can be efficiently tested given y and x' only (and the key e). The second is *privately verifiable GEOWFs*, where the relation $\mathcal{R}_e^{\mathcal{F}}(y, x')$ might not be efficiently testable given only $(y = f_e(x), x')$, but it is possible to efficiently test the relation given x in addition.

We note that standard EOWFs, as given in Definition 4, fall under the category of publicly verifiable GEOWFs, where the relation $\mathcal{R}_e^{\mathcal{F}}(y, x)$ simply tests whether $y = f_e(x)$.

DEFINITION 6 (GEOWFs). *An efficiently computable family of functions*

$$\mathcal{F} = \left\{ f_e : \{0,1\}^{\ell(n)} \to \{0,1\}^{\ell'(n)} \;\middle|\; e \in \{0,1\}^{m(n)}, n \in \mathbb{N} \right\} \enspace ,$$

associated with an efficient (probabilistic) key sampler $\mathcal{K}_{\mathcal{F}}$, is a GEOWF, with respect to a relation $\mathcal{R}_e^{\mathcal{F}}(y, x)$ on triples $(e, y, x) \in \{0,1\}^{m(n)+\ell'(n)+\ell(n)}$, if it is as follows:

1. $\mathcal{R}^{\mathcal{F}}$-Hard: *For any polynomial-size $\mathcal{A} = \{\mathcal{A}_n\}_{n \in \mathbb{N}}$ and security parameter $n \in \mathbb{N}$,*

$$\Pr_{\substack{e \leftarrow \mathcal{K}_{\mathcal{F}}(1^n) \\ x \leftarrow \{0,1\}^{\ell(n)}}} \left[\begin{array}{c} x' \leftarrow \mathcal{A}(e, f_e(x)) \\ \mathcal{R}_e^{\mathcal{F}}(f_e(x), x') = 1 \end{array} \right] \le \mathrm{negl}(n) \enspace .$$

2. $\mathcal{R}^{\mathcal{F}}$-Extractable: *For any PPT adversary $\mathcal{A}$, there exists a PPT extractor $\mathcal{E}$ such that for any polynomial b, security parameter $n \in \mathbb{N}$, and $z \in \{0,1\}^{b(n)}$,*

$$\Pr_{e \leftarrow \mathcal{K}_{\mathcal{F}}(1^n)} \left[\begin{array}{c} \exists x : f_e(x) = y \\ \mathcal{R}_e^{\mathcal{F}}(f_e(x), x') \neq 1 \end{array} \;\middle|\; \begin{array}{c} y \leftarrow \mathcal{A}(e; z) \\ x' \leftarrow \mathcal{E}(e; z) \end{array} \right] \le \mathrm{negl}(n) \enspace .$$

We further say that the function is

- Publicly verifiable *if there exists a polynomial-time tester $\mathcal{T}$ such that for any (x, x', e),*

$$\mathcal{R}_e^{\mathcal{F}}(f_e(x), x') = \mathcal{T}(e, f_e(x), x') \enspace ;$$

- Privately verifiable *if there exists a polynomial-time tester $\mathcal{T}$ such that for any (x, x', e),*

$$\mathcal{R}_e^{\mathcal{F}}(f_e(x), x') = \mathcal{T}(e, x, x') \enspace .$$

Bounded auxiliary input. The case of b-bounded auxiliary-input GEOWFs is defined analogously to b-bounded auxiliary-input-EOWFs. That is, $\mathcal{R}^{\mathcal{F}}$-hardness is defined exactly as in Definition 6, whereas $\mathcal{R}^{\mathcal{F}}$-hardness is only against adversaries with auxiliary input of an a priori fixed polynomial size $b(n)$.

Remark 3 (does $\mathcal{R}^{\mathcal{F}}$-hardness imply one-wayness). In principle, $\mathcal{R}^{\mathcal{F}}$-hardness may not imply one-wayness of $\mathcal{F}$. Although this is not needed for our purposes, we may further require that the relation $\mathcal{R}^{\mathcal{F}}$ include all pairs $(f_e(x), x)$ and thus ensure that $\mathcal{R}^{\mathcal{F}}$-hardness does imply one-wayness.

Remark 4 (GEOWFs versus proximity EOWFs). In [BCCT12], a different variant of EOWFs called *proximity EOWFs* is defined. There a proximity relation $\sim$ is defined on the range of the function. One-wayness is strengthened to require that not only is inverting $f_e(x)$ hard, but also finding x' such that $f_e(x) \sim f_e(x')$ is hard. Extractability is weakened so that the extractor is allowed to output x' as above, rather than an actual preimage. GEOWFs simply allow the relation to be even more general. In particular, any proximity EOWF with relation $\sim$ implies a GEOWF with relation $\mathcal{R}$, such that $\mathcal{R}(f_e(x), x') = 1$ iff $f_e(x) \sim f_e(x')$. Thus, the limitations we show in section 4 on GEOWFs apply to proximity EOWFs as well.

4. From IO to impossibility of unbounded-auxiliary-input EOWFs.

We show that if there exists IO, there do not exist (generalized) auxiliary-input EOWFs. We start by defining IO and puncturable PRFs.

4.1. Indistinguishability obfuscation.

IO was introduced in [BGI$^+$01] and given a first candidate construction in [GGH$^+$13b].

DEFINITION 7 (indistinguishability obfuscation [BGI$^+$01]). *A PPT algorithm $i\mathcal{O}$ is said to be an* indistinguishability obfuscator *INDO for a class of circuits $\mathcal{C}$ if it satisfies the following:*

1. Functionality: *For any $C \in \mathcal{C}$,*
$$\Pr_{i\mathcal{O}}\left[\forall x : i\mathcal{O}(C)(x) = C(x)\right] = 1 \ .$$
2. Indistinguishability: *For any class of circuit pairs $\{(C_n^{(1)}, C_n^{(2)}) \in \mathcal{C} \times \mathcal{C}\}_{n\in\mathbb{N}}$, where the two circuits $(C_n^{(1)}, C_n^{(2)})$ in each pair are of the same polynomial size $s(n) = n^{O(1)}$ and functionality, it holds that*
$$\left\{i\mathcal{O}(C_n^{(1)})\right\}_{n\in\mathbb{N}} \approx_c \left\{i\mathcal{O}(C_n^{(2)})\right\}_{n\in\mathbb{N}} \ .$$

4.2. Puncturable PRFs.

We next define puncturable PRFs. We consider a simple case of the puncturable PRFs where any PRF might be punctured at a single point. The definition is formulated as in [SW14] and, as shown in [BGI14, BW13, KPTZ13], is achieved by the GGM [GGM86] PRF.

DEFINITION 8 (puncturable PRFs). *Let ℓ, m be polynomially bounded length functions. An efficiently computable family of functions*
$$\mathcal{PRF} = \left\{\mathsf{PRF}_k : \{0,1\}^{m(n)} \to \{0,1\}^{\ell(n)} \;\middle|\; k \in \{0,1\}^n, n \in \mathbb{N}\right\} \ ,$$
associated with an efficient (probabilistic) key sampler $\mathcal{K}_{\mathcal{PRF}}$, is a puncturable PRF if there exists a polynomial-time puncturing algorithm Punc *that takes as input a key $k \in \{0,1\}^n$ and a point x^* and outputs a punctured key k_{x^*}, so that the following conditions are satisfied:*

1. Functionality is preserved under puncturing: *For every* $x^* \in \{0,1\}^{\ell(n)}$,

$$\Pr_{k \leftarrow \mathcal{K}_{\mathcal{PRF}}(1^n)}\left[\forall x \neq x^* : \mathsf{PRF}_k(x) = \mathsf{PRF}_{k_{x^*}}(x) \mid k_{x^*} = \mathsf{Punc}(k, x^*)\right] = 1 \ .$$

2. Indistinguishability at punctured points: *The following ensembles are computationally indistinguishable:*
   - $\{x^*, k_{x^*}, \mathsf{PRF}_k(x^*) \mid k \leftarrow \mathcal{K}_{\mathcal{PRF}}(1^n), k_{x^*} = \mathsf{Punc}(k, x^*)\}_{x^* \in \{0,1\}^{m(n)}, n \in \mathbb{N}}$,
   - $\{x^*, k_{x^*}, u \mid k \leftarrow \mathcal{K}_{\mathcal{PRF}}(1^n), k_{x^*}{=}\mathsf{Punc}(k, x^*),\ u \leftarrow \{0,1\}^{\ell(n)}\}_{x^* \in \{0,1\}^{m(n)}, n \in \mathbb{N}}$.

To be explicit, we include x^* in the distribution; throughout, we shall assume for simplicity that a punctured key k_{x^*} includes x^* in the clear.

4.3. The impossibility result. We now show that if indistinguishability obfuscators exist, there do not exist auxiliary-input EOWFs or GEOWFs according to Definitions 4 and 6.

Theorem 9. *Assuming IO for all circuits, neither EOWFs nor GEOWFs exist, with respect to common auxiliary input of unbounded polynomial length.*

Before proving the theorem we make two remarks.

Remark 5 (implications for other extractable primitives). GEOWFs are a minimal extractable cryptographic primitive in the sense that other extractable primitives such as ECRHs or succinct noninteractive arguments of knowledge (SNARKs) imply them. (For example, in [BCCT12], it is shown that SNARKs imply proximity ECRHs, which in turn imply proximity EOWFs, which as noted in Remark 4 imply GEOWFs.) These implications are invariant with respect to auxiliary input, and thus our limitation on common auxiliary input also holds with respect to these extractable primitives.

Remark 6 (auxiliary-input notions that are not ruled out). The limitation we prove relies critically on the adversary and extractor having *common* auxiliary input and does not apply if we only require extractability with respect to *individual* auxiliary input, as defined in Remark 2. The result does hold if we allow both individual and common auxiliary input. Also, our result does not apply for any distribution on common auxiliary inputs but rather shows that some specific auxiliary-input distribution fails extractability. In particular, we do not rule out natural distributions that may be conjectured to be "benign" (see Remark 2), such as the uniform distribution.

We now turn to prove Theorem 9. To prove the theorem, for any EOWF (respectively, GEOWF) family $\mathcal{F}$, we shall describe an adversary $\mathcal{A}$ and a distribution $\mathcal{Z}$ on auxiliary inputs, such that *any* extractor fails relative to auxiliary inputs sampled from $\mathcal{Z}$. For simplicity of exposition, we first concentrate on the case of plain EOWFs and then show how it directly extends to the case of GEOWFs.

4.3.1. The universal adversary. We consider a universal PPT adversary $\mathcal{A}$ that given $(e, z) \in \{0,1\}^{m(n)} \times \{0,1\}^{\mathrm{poly}(n)}$ parses z as a circuit and returns $z(e)$.

4.3.2. The auxiliary input distribution. Let $\mathcal{F}$ be a family of EOWFs and let $\mathcal{PRF}$ be a puncturable PRF family. We start by defining two families of circuits:

$$\mathcal{C} = \left\{ C_k : \{0,1\}^{m(n)} \to \{0,1\}^{\ell'(n)} \;\middle|\; k \in \{0,1\}^n, n \in \mathbb{N} \right\} \ ,$$

$$\mathcal{C}^* = \left\{ C_{k_{e^*}, y^*} : \{0,1\}^{m(n)} \to \{0,1\}^{\ell'(n)} \;\middle|\; k \in \{0,1\}^n, e^* \in \{0,1\}^{m(n)}, \right.$$
$$\left. y^* \in \{0,1\}^{\ell'}, n \in \mathbb{N} \right\} \ .$$

```
$C_k$
Hardwired: a PRF key $k \in \{0,1\}^n$.
Input: an EOWF key $e \in \{0,1\}^{m(n)}$.
    1. Compute $x = \mathsf{PRF}_k(e)$.
    2. Return $y = f_e(x)$.
```

FIG. 1. *The circuit C_k.*

```
$C_{k_{e^*},y^*}$
Hardwired: a punctured PRF key $k_{e^*} = \mathsf{Punc}(k, e^*)$ and $y^* \in \{0,1\}^{\ell'(n)}$.
Input: an EOWF key $e \in \{0,1\}^{m(n)}$.
    1. If $e \neq e^*$, compute $x = \mathsf{PRF}_{k_{e^*}}(e)$, and return $y = f_e(x)$.
    2. If $e = e^*$, return $y^*$.
```

FIG. 2. *The circuit $C_{k_{e^*},y^*}$.*

```
$Z_n$
  1. Sample $k \leftarrow \mathcal{K}_{\mathcal{PRF}}(1^n)$.
  2. Sample an obfuscation $z \leftarrow i\mathcal{O}([C_k]_s)$.
  3. Output $z$.
```

FIG. 3. *The auxiliary input distribution Z_n.*

Any circuit C_k, given a key e for an EOWF, applies PRF_k to e, obtains an input x, and returns the result of applying the EOWF f_e to x. The circuit is formally described in Figure 1.

The circuit $C_{k_{e^*},y^*}$ has a hardwired PRF key k_{e^*} that was derived from k by puncturing it at the point e^*. In addition, it has hardwired an output y^* to replace the punctured result. In particular, when $y^* = f_{e^*}(\mathsf{PRF}_k(e^*))$ the circuit $C_{k_{e^*},y^*}$ computes the same function as C_k. The circuit is formally described in Figure 2.

We are now ready to define our auxiliary-input distribution $\mathcal{Z} = \{Z_n\}_{n\in\mathbb{N}}$. Let $s = s(n)$ be the maximal size of circuits in either $\mathcal{C}$ or $\mathcal{C}^*$, corresponding to security parameter n, and denote by $[C]_s$ a circuit C padded with zeros to size s. Let $i\mathcal{O}$ be an indistinguishability obfuscator. The distribution Z_n simply consists of an obfuscated (padded) circuit C_k. The distribution is formally defined in Figure 3.

4.3.3. Adversary $\mathcal{A}$ does not have an extractor.

We next show that $\mathcal{A}$ cannot have any extractor $\mathcal{E}$ satisfying Definition 4. In fact, we show a stronger claim, namely, that for the auxiliary input distribution $\mathcal{Z}$, any extractor fails with overwhelming probability.

PROPOSITION 10. *Let $\mathcal{E}$ be any PPT candidate extractor for $\mathcal{A}$; then*

$$\Pr_{\substack{e \leftarrow \mathcal{K}_{\mathcal{F}}(1^n)\\ z \leftarrow Z_n}}\left[\begin{array}{c} \exists x : f_e(x) = y \\ f_e(x') \neq y \end{array} \middle| \begin{array}{c} y \leftarrow \mathcal{A}(e;z) \\ x' \leftarrow \mathcal{E}(e;z) \end{array}\right] \geq 1 - \mathrm{negl}(n) \ .$$

We note that since the key e is sampled above independently of the auxiliary input z, the above indeed disproves extractability.

Proof of Proposition 10. First, we note that

$$\Pr_{\substack{e\leftarrow\mathcal{K}_{\mathcal{F}}(1^n)\\ z\leftarrow Z_n}}\left[\begin{array}{c} y\leftarrow\mathcal{A}(e;z)\\ \exists x: f_e(x)=y\end{array}\right]=1\ ;$$

indeed, by the definition of $\mathcal{A}$ and Z_n, and the correctness of $i\mathcal{O}$,

$$\mathcal{A}(e,z)=z(e)=C_k(e)=f_e(\mathsf{PRF}_k(e))\ ,$$

where $C_k\in\mathcal{C}$ is the circuit obfuscated in z, i.e., $z=i\mathcal{O}([C_k]_s)$.

Now, assume toward contradiction that, for infinitely many $n\in\mathbb{N}$, the extractor $\mathcal{E}$ successfully outputs a preimage with noticeable probability $\varepsilon(n)$, i.e.,

$$\Pr_{\substack{e\leftarrow\mathcal{K}_{\mathcal{F}}(1^n)\\ z\leftarrow Z_n}}\left[\begin{array}{c} x'\leftarrow\mathcal{E}(e;z)\\ f_e(x')=z(e)=f_e(\mathsf{PRF}_k(e))\end{array}\right]\geq\varepsilon(n)\ ,$$

where as before, $z=i\mathcal{O}([C_k]_s)$.

Then, for every e^* we consider an alternative distribution $Z_n(e^*,y^*)$ that instead of sampling a circuit C_k samples a circuit $C_{k_{e^*},y^*}$, by first sampling k as usual, and then computing $y^*=f_{e^*}(\mathsf{PRF}_k(e^*))$, and the punctured key k_{e^*}. (Note that $Z_n(e^*,y^*)$ is actually only parameterized by e^*; we add y^* to the notation to be more explicit.) We claim that the extractor still succeeds in finding a preimage, i.e.,

$$\Pr_{\substack{e^*\leftarrow\mathcal{K}_{\mathcal{F}}(1^n)\\ z^*\leftarrow Z_n(e^*,y^*)}}\left[\begin{array}{c} x'\leftarrow\mathcal{E}(e^*;z^*)\\ f_{e^*}(x')=z^*(e^*)=y^*=f_{e^*}(\mathsf{PRF}_k(e^*))\end{array}\right]\geq\varepsilon(n)-\mathrm{negl}(n)\ .$$

This follows from the $i\mathcal{O}$ indistinguishability guarantee, since for any e^* and k, C_k and $C_{k_{e^*},y^*}$ compute the same function.

Next, we consider another experiment where $Z_n(e^*,y^*)$ is altered to a new distribution $Z_n(e^*,r)$ that, instead of sampling $y^*=f_{e^*}(\mathsf{PRF}_k(e^*))$ in $C_{k_{e^*},y^*}$, samples $y^*=f_{e^*}(r)$, for an independent random $r\leftarrow\{0,1\}^{\ell}$. We claim that

$$\Pr_{\substack{e^*\leftarrow\mathcal{K}_{\mathcal{F}}(1^n)\\ z^*\leftarrow Z_n(e^*,r)}}\left[\begin{array}{c} x'\leftarrow\mathcal{E}(e^*;z^*)\\ f_{e^*}(x')=z^*(e^*)=y^*=f_{e^*}(r)\end{array}\right]\geq\varepsilon(n)-\mathrm{negl}(n)\ ;$$

indeed, this follows from the fact that $\mathsf{PRF}_k(e^*)$ is pseudorandom, even given the punctured key k_{e^*} (as guaranteed by Definition 8).

To complete the proof, we deduce that $\mathcal{E}$ can be used to break the one-wayness of $\mathcal{F}$. Indeed, given a random key e^*, and a challenge $y^*=f_{e^*}(r)$, an inverter can simply sample a punctured k_{e^*} on its own, construct the circuit $C_{k_{e^*},y^*}$, with its challenge y^* hardwired in, and sample an obfuscation $z^*\leftarrow i\mathcal{O}(C_{k_{e^*},y^*})$. Finally, it runs $\mathcal{E}(e^*,z^*)$ to invert y^*, with the same probability $\varepsilon(n)-\mathrm{negl}(n)$. □

Extending the result to GEOWFs. The result directly extends to show that no $\mathcal{F}$ can even be a GEOWF with respect to auxiliary input and any relation $\mathcal{R}^{\mathcal{F}}$. Concretely, we would consider the exact same universal adversary and auxiliary-input distribution $\mathcal{Z}$. The proof goes along the same lines: instead of an extractor that finds a preimage x of $y=z(e)$, we start from an extractor that finds $x\in\mathcal{R}_e^{\mathcal{F}}(y)$. Then, instead of obtaining an inverter that breaks the one-wayness of $\mathcal{F}$, we obtain an inverter that breaks the $\mathcal{R}^{\mathcal{F}}$-hardness of $\mathcal{F}$. The proof follows the exact same arguments. The only thing that should be noted is that when invoking the indistinguishability given by $i\mathcal{O}$, in the first hybrid, and then the indistinguishability given by pseudorandomness at punctured points, in the second, it can indeed be efficiently tested whether the extractor successfully obtained a witness $x \in \mathcal{R}_e^{\mathcal{F}}(y)$. This is clear in the case that $\mathcal{R}^{\mathcal{F}}$ is publicly verifiable and is also true in the case that it is privately verifiable, as in both cases y is computed directly from a pre image ($\mathsf{PRF}_k(e^*)$, in the first, and r, in the second) that is known to the distinguisher, and which allows testing the relation.

Finally, to deduce Theorem 9, we note that puncturable PRFs can be constructed from one-way functions. Furthermore, EOWF is already an OWF, and any GEOWF with $\mathcal{R}^{\mathcal{F}}$-hardness implies that NP $\neq$ coRP, which in conjunction with $i\mathcal{O}$ implies OWFs [KMN+14]. Thus, the impossibility of auxiliary-input EOWFs and GEOWFs is implied by IO without any further assumptions.

5. Bounded-auxiliary-input extractable one-way functions.

In this section, we construct bounded-auxiliary-input EOWFs and GEOWFs. Before presenting the construction, we define *universal delegation for deterministic computations*, which is the main tool we rely on. The notion we consider is a slight enhancement of standard delegation schemes in the spirit of Barak and Goldreich's [BG08] *universal arguments for nondeterministic computations*. We show how to instantiate this notion based on the delegation scheme of Kalai, Raz, and Rothblum [KRR14], which follows the standard notion of delegation.

5.1. Universal delegation for deterministic computations.

In what follows, we denote by $\mathcal{L}_{\mathcal{U}}$ the universal language consisting of all tuples $(\mathcal{M}, x, t)$ such that $\mathcal{M}$ accepts x within t steps. We denote by $\mathcal{L}_{\mathcal{U}}(T)$ all pairs $(\mathcal{M}, x)$ such that $(\mathcal{M}, x, T) \in \mathcal{L}_{\mathcal{U}}$.

Let $T(n) \in (2^{\omega(\log n)}, 2^{\mathrm{poly}(n)})$ be a computable superpolynomial function. A universal delegation system for Dtime(T) consists of three algorithms $(\mathcal{G}, \mathcal{P}, \mathcal{V})$ that work as follows:

- The (probabilistic) generator $\mathcal{G}$, given a security parameter 1^n, outputs a *reference string* σ and a corresponding *verification state* τ; in particular, $\mathcal{G}$ is independent of any statement to be proven later.
- The honest prover $\mathcal{P}(\mathcal{M}, x; \sigma)$ produces a certificate π for the fact that $(\mathcal{M}, x) \in \mathcal{L}_{\mathcal{U}}(T(n))$.
- The verifier $\mathcal{V}(\mathcal{M}, x; \pi, \tau)$ verifies the validity of π.

Formally, a universal delegation system is defined as follows.

Definition 11 (universal delegation). *A triple $(\mathcal{G}, \mathcal{P}, \mathcal{V})$ is a noninteractive universal argument system for for Dtime(T) if it satisfies the following:*

1. Perfect completeness: *For any $n \in \mathbb{N}$ and $(\mathcal{M}, x) \in \mathcal{L}_{\mathcal{U}}(T(n))$,*

$$\Pr\left[\mathcal{V}(\mathcal{M}, x; \pi, \tau) = 1 \;\middle|\; \begin{array}{c} (\sigma, \tau) \leftarrow \mathcal{G}(1^n) \\ \pi \leftarrow \mathcal{P}(\mathcal{M}, x; \sigma) \end{array}\right] = 1 \ .$$

2. Adaptive soundness for a bounded number of statements: *There is a polynomial b such that for any polynomial-size prover $\mathcal{P}^*$, $n \in \mathbb{N}$, and set of at most $2^{b(n)}$ false statements $S \subseteq \{0,1\}^{\mathrm{poly}(n)} \setminus \mathcal{L}_{\mathcal{U}}(T(n))$,*

$$\Pr\left[\mathcal{V}(\mathcal{M}, x; \pi, \tau) = 1 \;\middle|\; \begin{array}{r} (\sigma, \tau) \leftarrow \mathcal{G}(1^n) \\ (\mathcal{M}, x, \pi) \leftarrow \mathcal{P}^*(\sigma) \\ (\mathcal{M}, x) \in S \end{array}\right] \leq \mathrm{negl}(n) \ .$$

3. Fast verification and relative prover efficiency: *There exists a polynomial p such that for every $n \in \mathbb{N}$, $t \leq T(n)$, and $(\mathcal{M}, x) \in \mathcal{L}_{\mathcal{U}}(t)$,*

- *the generator $\mathcal{G}$ runs in time $p(n)$;*
- *the verifier $\mathcal{V}$ runs in time $p(n + |\mathcal{M}| + |x|)$;*
- *the prover $\mathcal{P}$ runs in time $p(n + |\mathcal{M}| + |x| + t)$.*

The system is said to be publicly verifiable *if soundness is maintained when the malicious prover is also given the verification state τ. In this case, we will assume w.l.o.g. that the verification state τ appears in the clear in the reference string σ.*

Existence and connection to standard delegation of computation. There are two differences between the standard notion of delegation for deterministic computations (see, e.g., [KRR14]) and the universal delegation notion defined above:

1. A delegation system is associated with a given language $\mathcal{L}(\mathcal{M})$ for a fixed deterministic machine $\mathcal{M}$, and the corresponding efficiency parameters depend on the worst-case running time $T_{\mathcal{M}}$ of $\mathcal{M}$. In particular, the generator $\mathcal{G}$ depends on $T_{\mathcal{M}}$ as an extra parameter, and the prover's efficiency is polynomial in the worst-case running time $T_{\mathcal{M}}$.
2. In basic delegation schemes only nonadaptive soundness is guaranteed; in particular, the generator's message σ may, in principle, depend on the input x.

Kalai, Raz, and Rothblum [KRR14] show how to construct such a privately verifiable *delegation scheme* for every language in $\text{Dtime}(T) \subseteq \text{EXP}$, assuming subexponentially secure private information retrieval schemes, which can in turn be constructed based on the subexponential learning with errors assumption [BV11].

In order to get a (privately verifiable) universal delegation for $\text{Dtime}(T)$, we could potentially use their result with respect to a universal machine and worst-case running time $O(T)$. However, this solution would lack the required prover efficiency, as the prover will always run in time $\text{poly}(T)$, even for machines $\mathcal{M}$ with running time $t_{\mathcal{M}} << T$. This is undesired in our case, as we will be interested in T that is superpolynomial. Fortunately, a rather standard transformation does allow us to get the required efficiency from their result. Specifically, we could run the generator in their solution to generate a reference string and verification state (σ, τ) for computations of size $t \; \forall \; t \in \left\{1, 2, 2^2, \dots, 2^{\log T}\right\}$, and have the prover and verifier use the right (σ, τ) according to the concrete running time $t_{\mathcal{M}} < T$, guaranteeing that the prover's running time is at most $\text{poly}(2t_{\mathcal{M}})$ as required.

As for adaptivity, in their scheme, the generator does work independently of the input x, but only nonadaptive soundness is shown; namely, soundness is guaranteed only when σ is generated independently of x. To guarantee soundness for adaptively chosen inputs x from a set S of size at most $2^{b(n)}$, we may repeat the above argument $O(b(n)) + n$ times. Assuming that the underlying delegation scheme is secure against provers that run in time $2^{O(b(n))}$ (by choosing the security parameter in the [KRR14] scheme appropriately), the parallel repetition exponentially reduces the soundness error to $2^{-b(n)-n}$ (see, e.g., [BIN97]). Taking a union bound over all $2^{b(n)}$ adaptive choices of x yields the required soundness. The $O(b(n))$-factor hit in succinctness and verification time are still tolerable for our purposes (and still satisfy the above definition).

THEOREM 12 (follows from [KRR14]). *Assuming the learning with errors problem is subexponentially hard for any $b(n) = \text{poly}(n)$, and $T(n) \in (2^{\omega(\log n)}, 2^{\text{poly}(n)})$, there exists a (privately verifiable) universal delegation scheme with adaptive soundness for $2^{b(n)}$ statements.*

5.2. Pseudorandom generators.

We recall the definition of a cryptographic PRG.

```
Inputs: $(s, r, \mathsf{pad})$ of respective lengths $(n, n, \ell(n) - 2n)$.
    1. Sample reference string and verification state $(\sigma, \tau) \leftarrow \mathcal{G}(1^n; r)$ for universal delegation.
    2. Compute $v = \mathsf{PRG}(s)$.
    3. Output $(\sigma, v)$.
```

FIG. 4. *The function f_n.*

DEFINITION 13. *A polynomial-time function* PRG *stretching n bits to $\ell(n) > n$ bits is a* PRG *if its output is computationally indistinguishable from a truly random one:*

$$\{\mathsf{PRG}(s) \mid s \leftarrow \{0,1\}^n\}_{n\in\mathbb{N}} \approx_c \left\{u \;\middle|\; u \leftarrow \{0,1\}^{\ell(n)}\right\}_{n\in\mathbb{N}} .$$

5.3. Constructions. We now present our constructions of bounded-auxiliary-input EOWFs and GEOWFs. We start with the construction of GEOWFs, based on any universal delegation scheme. We then give a construction of the standard (rather than generalized) EOWFs based on publicly verifiable universal delegation schemes with an additional key validation property (satisfied by existing candidates).

5.3.1. The generalized extractable one-way function. In what follows,

- let $b(n)$ be a polynomial,
- let PRG be a pseudorandom generator stretching n bits to $b(n) + n$ bits,
- and let $(\mathcal{G}, \mathcal{P}, \mathcal{V})$ be a universal delegation scheme for Dtime$(T(n))$ for some function $T(n) \in (2^{\omega(\log n)}, 2^{\text{poly}(n)})$, with adaptive soundness for $2^{b(n)}$ statements. We assume that the system handles statements of the form $(\mathcal{M}, v) \in \{0,1\}^{b(n)} \times \{0,1\}^{b(n)+n}$ asserting that "$\mathcal{M}(1^n)$ outputs v in $T(n)$ steps." Assume that $\mathcal{G}(1^n; r)$ uses randomness of size n to output a reference string of polynomial size $m(n)$, and a verification state τ (if the system is publicly verifiable, then τ appears in σ). Assume that $\mathcal{P}$ outputs certificates π of size $p(n)$.

We construct a keyless family of functions $\mathcal{F} = \{f_n\}_{n\in\mathbb{N}}$, consisting of one function $f_n : \{0,1\}^{\ell(n)} \to \{0,1\}^{\ell'(n)}$, for each security parameter n, where $\ell(n) = \max(2n, b(n) + p(n))$ and $\ell'(n) = m(n) + b(n) + n$. The function is given in Figure 4 and is followed by the corresponding relation $\mathcal{R}^{\mathcal{F}}$.

We now define the corresponding relation $\mathcal{R}^{\mathcal{F}} = \left\{\mathcal{R}_n^{\mathcal{F}}\right\}_{n\in\mathbb{N}}$ in Figure 5, which will be publicly verifiable (respectively, privately verifiable) if the universal delegation scheme is publicly (respectively, privately) verifiable. For simplicity, we assume that the universal delegation scheme is such that for every valid reference string σ produced by $\mathcal{G}$, there is a single possible verification state τ (this can always be achieved by adding a commitment to τ inside σ).

CLAIM 1. $\mathcal{R}^{\mathcal{F}}$ *is publicly verifiable (respectively, privately verifiable) if* $(\mathcal{G}, \mathcal{P}, \mathcal{V})$ *is publicly verifiable (respectively, privately verifiable).*

Proof. First, by definition, when $(\mathcal{G}, \mathcal{P}, \mathcal{V})$ is publicly verifiable, τ can be obtained from σ, verification can be done efficiently, and thus the relation $\mathcal{R}_n^{\mathcal{F}}$ can be efficiently tested given (σ, v, π).

Next, assume that $(\mathcal{G}, \mathcal{P}, \mathcal{V})$ is privately verifiable. Recall that showing that $\mathcal{R}_n^{\mathcal{F}}$ is privately verifiable means that given any preimage x such that $y = f_n(x)$, we can efficiently test $\mathcal{R}_n^{\mathcal{F}}(y, x')$. Indeed, given such a preimage $x = (s, r, \mathsf{pad})$, we can obtain

```
Inputs:
$y = f_n(x) = (\sigma, v)$ of respective lengths $(m(n), b(n) + n)$,
$x' = (\mathcal{M}, \pi, \mathsf{pad})$ of respective lengths $(b(n), p(n), \ell(n) - b(n) - p(n))$.
    1. Find the (unique) verification state $\tau$ corresponding to the reference string $\sigma$:
    2. Run $\mathcal{V}(\mathcal{M}, v; \pi, \tau)$ to verify the statement "$\mathcal{M}(1^n)$ outputs $v$ in $T(n)$ steps".
    3. Return 1 iff verification passes.
```

FIG. 5. *The relation* $\mathcal{R}^{\mathcal{F}}_n(f_n(x), x')$.

the generator randomness r and run $\mathcal{G}(1^n; r)$ to obtain the (unique) verification state τ corresponding to σ and efficiently test $\mathcal{R}^{\mathcal{F}}_n$. □

Remark 7 (one-wayness versus $\mathcal{R}^{\mathcal{F}}$-hardness of $\mathcal{F}$). The relation $\mathcal{R}^{\mathcal{F}}$ defined above is such that $(f_n(x), x)$ may not satisfy the relation. In particular, this means that $\mathcal{R}^{\mathcal{F}}$-hardness may not imply one-wayness of $\mathcal{F}$. While this is not needed for our purposes, the relation $\mathcal{R}^{\mathcal{F}}$ can be augmented to also include all pairs $(f_n(x), x)$, and $\mathcal{R}^{\mathcal{F}}$-hardness will still be preserved; that is, the function we define is one-way in the usual sense.

We now turn to show that $\mathcal{F}$ is a GEOWF with respect to $\mathcal{R}^{\mathcal{F}}$.

THEOREM 14. *The function family $\mathcal{F} = \{f_n\}_{n\in\mathbb{N}}$, given in Figure* 4*, is a GEOWF, with respect to $\mathcal{R}^{\mathcal{F}}$, against $(b(n) - \omega(1))$-bounded auxiliary input.*

High-level idea behind the proof. To see that $\mathcal{F}$ is $\mathcal{R}^{\mathcal{F}}$-hard, note that to break $\mathcal{R}^{\mathcal{F}}$-hardness, an adversary given a random image (σ, v), where $v = \mathsf{PRG}(s)$ is of length $b(n) + n$, has to come up with a "small" machine $\mathcal{M}$, whose description length is at most $b(n)$, and a proof that $\mathcal{M}$ outputs v (within a $T(n)$ steps). However, in an indistinguishable world where v is a truly random string, v would almost surely have high Kolomogorov complexity, and a short machine $\mathcal{M}$ that outputs v would not exist. Thus, in this case, the breaker has to produce an accepting proof for a false statement and violate the soundness of the universal delegation scheme.

As for extraction, given a polynomial-time machine $\mathcal{M}_z$ with short advice z that outputs (σ, v), where σ is a valid reference string for the universal delegation scheme, the extractor simply computes a proof π for the fact that $\mathcal{M}_z$ outputs v, and outputs the witness $(\mathcal{M}_z, \pi; \mathsf{pad})$. By the completeness of the universal delegation scheme, the proof π is indeed accepting, and the witness satisfies $\mathcal{R}^{\mathcal{F}}$. Furthermore, by the relative prover efficiency of the universal delegation scheme, the extractor runs in time that is polynomial in the running time of the adversary $\mathcal{M}_z$.

Proof of Theorem 14. We first show $\mathcal{R}^{\mathcal{F}}$-hardness and then show $\mathcal{R}^{\mathcal{F}}$-extractability.

$\mathcal{R}^{\mathcal{F}}$-hardness. Assume there exists a breaker $\mathcal{B}$ that, given $y = (\sigma, v)$, where $\sigma \leftarrow \mathcal{G}(1^n)$, and $v \leftarrow \mathsf{PRG}(U_n)$, finds $x = (\mathcal{M}, \pi, \mathsf{pad})$ such that $\mathcal{R}^{\mathcal{F}}_n(y, x) = 1$ with noticeable probability $\varepsilon(n)$. We construct a prover $\mathcal{P}^*$ that breaks the adaptive soundness of the universal delegation scheme (for $2^{b(n)}$ statements), with probability $\varepsilon(n) - \text{negl}(n)$. $\mathcal{P}^*$, given σ, first samples on its own $\tilde{v} \leftarrow U_{b(n)+n}$ (independently of σ), and then runs $\mathcal{B}(\sigma, \tilde{v})$ to obtain a machine $\mathcal{M}$ of size $b(n)$, and a proof π.

We first claim that with probability $\varepsilon(n) - \text{negl}(n)$, π is an accepting proof for the statement $(\mathcal{M}, \tilde{v})$ asserting that "$\mathcal{M}(1^n)$ outputs $\tilde{v}$ in $T(n)$ steps." Indeed, the view of $\mathcal{B}$ in the above experiment is identical to its real view, except that it gets a truly random $\tilde{v}$, rather than a pseudorandom v that was generated using PRG. Thus, the claim follows by the PRG guarantee.

Next, we note that since $\tilde{v}$ is a $(b(n)+n)$-long random string, except with negligible probability 2^{-n}, there does not exist $\mathcal{M}$ of size $b(n)$ that outputs $\tilde{v}$. Thus, $\mathcal{P}^*$ produces an accepting proof for one of $2^{b(n)}$ false statements given by the adaptive choice of $\mathcal{M} \in \{0,1\}^{b(n)}$ and violates the soundness of the universal delegation scheme.

$\mathcal{R}^{\mathcal{F}}$-extractability. We now show $\mathcal{R}^{\mathcal{F}}$-extractability. We, in fact, show that there is one universal PPT extractor $\mathcal{E}$ that can handle any PPT adversary $\mathcal{M}$ with advice of size at most $b(n) - \omega(1)$. In what follows, for adversarial code $\mathcal{M}$ and advice $z \in \{0,1\}^{b(n)-\omega(1)}$, denote by $\mathcal{M}_z$ the machine that, on input 1^n, runs $\mathcal{M}(1^n; z)$. Assume that $\mathcal{M}_z$ has description size at most $b(n)$ and that, on input 1^n, after at most $t_{\mathcal{M}} < T(n)$ steps it outputs

$$\mathcal{M}_z(1^n) = y = (\sigma, v) \in \mathsf{Image}(f_n) \ .$$

The extractor $\mathcal{E}(\mathcal{M}, z, 1^{t_{\mathcal{M}}})$ performs the following:

1. Constructs $\mathcal{M}_z$.
2. Computes a certificate π for the fact that "$\mathcal{M}_z(1^n) = v$."
3. Outputs $x' = (\mathcal{M}_z, \pi, \mathsf{pad})$.

The fact that $x' \in \mathcal{R}^{\mathcal{F}}(y)$ follows directly from the perfect completeness of the universal delegation scheme. Furthermore, by the relative prover efficiency of the universal delegation scheme, the extractor runs in time that is polynomial in the running time $t_{\mathcal{M}}$ of the adversary. □

Remark 8 ($\mathcal{R}^{\mathcal{F}}$-hardness against superpolynomial adversaries). In section 6.4.2, we shall require GEOWFs that are $\mathcal{R}^{\mathcal{F}}$-hard even against adversaries of size $\mathrm{poly}(T(n))$ for some superpolynomial function $T(n)$. Such GEOWFs can be obtained from the above construction by using a PRG that is secure against $\mathrm{poly}(T(n))$ adversaries and a universal delegation scheme that is sound against such adversaries (such a universal delegation scheme can be obtained from [KRR14], based on an appropriately strong private information retrieval scheme).

5.3.2. The standard extractable one-way function. We construct a standard EOWF based on publicly verifiable universal delegation schemes that have an additional property that says that, in addition to perfect completeness for an honestly chosen reference string σ (which in the publicly verifiable case is also the verification state), it is also possible to check whether any given σ is valid or more generally admits perfect completeness. We note that existing candidates for publicly verifiable universal delegation schemes indeed have this property.[7]

DEFINITION 15 (universal delegation scheme with key validation). *A publicly verifiable universal delegation scheme is said to have key validation if there exists a polynomial-time algorithm* Valid*, such that for any* $\sigma \in \{0,1\}^{m(n)}$*, if* $\mathsf{Valid}(\sigma) = 1$*, then the system has perfect completeness with respect to* σ*. That is, proofs for true statements, generated and verified using* σ*, are always accepted.*

We now turn to describe the construction. At a very high level the idea behind the construction is to embed the structure of the previous GEOWF function and relation into a standard EOWF.

[7] Indeed, in Micali's CS proofs [Mic00], perfect completeness holds with respect to all possible keys for a hash function. In the publicly verifiable instantiations of the SNARKs from [BCCT13] it is possible to verify the validity of σ using a bilinear map.

```
Inputs: $(i, (s, r), (\sigma, \mathcal{M}, v, \pi))$ of respective lengths $(n, (n, n), (m(n), b(n), b(n) + n, p(n)))$
    • If $i \notin \{0^n, 1^n\}$:
        1. Compute $v^* = \mathsf{PRG}(s)$.
        2. Sample a reference string $\sigma^* \leftarrow \mathcal{G}(1^n; r)$.
        3. Output $(\sigma^*, v^*)$.
    • If $i = 0^n$:
        1. Perform the following tests:
            – Run $\mathsf{Valid}(\sigma)$ to check the validity of $\sigma$,
            – Run $\mathcal{V}(\mathcal{M}, v; \pi, \sigma)$ to verify the statement "$\mathcal{M}(1^n)$ outputs $v$ in $T(n)$ steps",
           If both accept, output $(\sigma, v)$.
        2. Otherwise, output $\perp$.
    • If $i = 1^n$, output $\perp$.
```

Fig. 6. *The function f_n.*

- Let $b(n)$ be a polynomial.
- Let PRG be a pseudorandom generator stretching n bits to $b(n) + n$ bits.
- Let $(\mathcal{G}, \mathcal{P}, \mathcal{V})$ be a universal delegation scheme with the same parameters as in the above GEOWF construction and with the additional key validation property.

We construct a keyless family of functions $\mathcal{F} = \{f_n\}_{n\in\mathbb{N}}$, consisting of one function $f_n : \{0,1\}^{\ell(n)} \to \{0,1\}^{\ell'(n)}$, for each security parameter n, where $\ell(n) = 4n + 2b(n) + m(n) + p(n)$ and $\ell'(n) = m(n) + b(n) + n$. The function is given in Figure 6.

We now turn to show that $\mathcal{F}$ is an EOWF.

Theorem 16. *The function family $\mathcal{F} = \{f_n\}_{n\in\mathbb{N}}$, given in Figure* 6*, is an EOWF, against* $(b(n) - \omega(1))$*-bounded auxiliary-input.*

High-level idea behind the proof. To see that $\mathcal{F}$ is one-way, note that, except with negligible probability, a random image comes from the "normal branch of the function," where $i \notin \{0^n, 1^n\}$, and includes an honestly sampled σ and a pseudorandom string $v = \mathsf{PRG}(s)$. To invert it, an adversary must either invert $\mathsf{PRG}(s)$, allowing it to produce a "normal branch" preimage, or obtain a short machine $\mathcal{M}$ and an accepting proof π that $\mathcal{M}$ outputs v, allowing it to produce a "trapdoor branch" preimage. In the first case, the inverter violates the one-wayness of PRG. In the second case, the inverter can be used to break the soundness of the universal delegation scheme as in the proof of Theorem 14 (leveraging the fact that a truly random $\tilde{v}$ almost surely cannot be computed by a short machine).

As for extraction, given a polynomial-time machine $\mathcal{M}_z$ with short advice z that outputs $(\sigma, v) \neq \perp$, by the definition of f_n, σ is a valid reference string for the universal delegation scheme (indeed, $\perp$ is an image that indicates an improper reference string σ, or a nonaccepting proof π). In this case, the extractor simply computes a proof π for the fact that $\mathcal{M}_z$ outputs v, and outputs the preimage $(0^n, (0^n, 0^n), (\sigma, \mathcal{M}_z, v, \pi))$. By the completeness of the universal delegation scheme, for a valid σ, the proof π is indeed accepting. By the relative prover efficiency of the universal delegation scheme, the extractor runs in time that is polynomial in the running time of the adversary $\mathcal{M}_z$. The only other case to consider is where $\mathcal{M}_z$ outputs $\perp$, in which case producing a preimage is easily done by setting $i = 1^n$.

Proof of Theorem 16. We first show one-wayness and then show extractability.

One-wayness. Assume there exists an inverter $\mathcal{I}$ that, given $y = f_n(x)$, where $x \leftarrow U_{\ell(n)}$, finds a preimage $x' = (i', (s', r'), (\sigma', \mathcal{M}', v', \pi'))$ with noticeable probability $\varepsilon(n)$. We construct a prover $\mathcal{P}^*$ that breaks the adaptive soundness of the universal delegation scheme (for $2^{b(n)}$ statements), with probability $\varepsilon(n) - \text{negl}(n)$. $\mathcal{P}^*$ is defined as in the proof of Theorem 14: given σ, it first samples on its own $\tilde{v} \leftarrow U_{b(n)+n}$ (independently of σ), and then runs $\mathcal{I}(\sigma, \tilde{v})$ to obtain $x' = (i', (s', r'), (\sigma', \mathcal{M}', v', \pi'))$.

CLAIM 2. *With probability $\varepsilon(n) - \text{negl}(n)$, π' is an accepting proof, with respect to σ, for the statement $(\mathcal{M}', v)$, attesting that "$\mathcal{M}'(1^n)$ outputs $\tilde{v}$ in $T(n)$ steps."*

The claim will conclude the proof of one-wayness. Indeed, as in the proof of Theorem 14, except with negligible probability, there does not exist a machine $\mathcal{M}'$ of size $b(n)$ that outputs $\tilde{v}$, which is a $(b(n) + n)$-long random string. This means that $\mathcal{I}$ outputs an accepting proof for one of $2^{b(n)}$ false statements (given different $\mathcal{M}' \in \{0, 1\}^{b(n)}$) and violates the soundness of the universal delegation scheme.

Proof of Claim 2. To prove the claim, we first consider a hybrid experiment where $\mathcal{I}$ samples a pseudorandom $v \leftarrow \mathsf{PRG}(U_n)$ instead of a truly random $\tilde{v}$. By the PRG guarantee, we know that the probability of outputting $(\mathcal{M}', \pi)$ as required by the claim changes at most by a neglible amount $\text{negl}(n)$. Next, we note that the view of $\mathcal{I}$ in the hybrid experiment is identical to its view in the real world where it receives a random image $y = (\sigma, v)$. Furthermore, whenever $\mathcal{I}$ finds a preimage $x' = (i', (s', r'), (\sigma', \mathcal{M}', v', \pi'))$ of y such that $i' = 0^n$, by the definition of f_n, $(\sigma', v') = (\sigma, v)$, and π' must be an accepting proof for the statement $(\mathcal{M}', v' = v)$, with respect to $\sigma' = \sigma$.

Since we know that $\mathcal{I}$ inverts the function with probability $\varepsilon(n)$, it thus suffices to show that the preimage it finds is such that $i = 0^n$, except with negligible probability. Indeed, whenever $\mathcal{I}$ finds a preimage such that $i' \notin \{0^n, 1^n\}$, by the definition of f_n, it inverts $v = \mathsf{PRG}(s)$, contradicting the one-wayness of PRG. Also, a preimage of (σ, v) cannot have $i' = 1^n$, assuming $(\sigma, v) \neq \perp$, which is the case with overwhelming probability. This concludes the proof of the claim. ◻

Extractability. We show that there is one universal PPT extractor $\mathcal{E}$ that can handle any PPT adversary $\mathcal{M}$ with advice of size at most $b(n) - \omega(1)$. The proof is similar to the extractability proof of Theorem 14. In what follows, for adversarial code $\mathcal{M}$ and advice $z \in \{0, 1\}^{b(n) - \omega(1)}$, denote by $\mathcal{M}_z$ the machine that, on input 1^n, runs $\mathcal{M}(1^n; z)$. Assume that $\mathcal{M}_z$ has description size at most $b(n)$ and that, on input 1^n, after at most $t_{\mathcal{M}} < T(n)$ steps it outputs

$$\mathcal{M}_z(1^n) = y = (\sigma, v) \in \mathsf{Image}(f_n) \ .$$

The extractor $\mathcal{E}(\mathcal{M}, z, 1^{t_{\mathcal{M}}})$ performs the following:

1. It computes $(\sigma, v) = \mathcal{M}_z(1^n)$.
2. If $(\sigma, v) \neq \perp$, the extractor
   (a) constructs $\mathcal{M}_z$,
   (b) computes a certificate π for the fact that "$\mathcal{M}_z(1^n) = v$,"
   (c) outputs $x' = (0^n, (0^n, 0^n), (\sigma, \mathcal{M}_z, v, \pi))$.
3. If $(\sigma, v) = \perp$, the extractor outputs $x' = (1^n, (0^n, 0^n), (0^{m(n)}, 0^{b(n)}, 0^{b(n)+n}, 0^{p(n)}))$.

If $(\sigma, v) \neq \perp$, we know that σ must be valid. In this case, the extractor's output $x' = (0^n, (0^n, 0^n), (\sigma, \mathcal{M}_z, v, \pi))$ is a valid preimage due to the perfect completeness of the universal delegation scheme.

If $(\sigma, v) = \perp$, the extractor's output $x' = (1^n, (0^n, 0^n), (0^{m(n)}, 0^{b(n)}, 0^{b(n)+n}, 0^{p(n)}))$ is a preimage by definition.

Finally, we note that by the relative prover efficiency of the universal delegation scheme, the extractor runs in time that is polynomial in the running time $t_{\mathcal{M}}$ of the adversary . □

6. 2-Message and 3-message ZK against bounded-auxiliary-input verifiers.

In this section, we define and construct 2- and 3-message ZK arguments against verifiers with bounded auxiliary input, based on GEOWFs. We start by presenting the definition of such ZK arguments and two tools which will be of use. Then, we move on to describe our constructions.

6.1. Definition.

The standard definition of ZK [GMR89, Gol04] considers adversarial verifiers with nonuniform auxiliary input of arbitrary polynomial size. We consider a relaxed notion of ZK against verifiers that have bounded nonuniform advice but arbitrary polynomial running time. This relaxed notion, in particular, includes ZK against uniform verifiers (sometimes referred to as *plain ZK* [BLV06]).

Concretely, we shall focus on PPT verifiers V^* having advice z of an a priori bounded size $b(n)$, and using an arbitrary polynomial number of random coins.

DEFINITION 17. *An argument system (P, V) for an NP relation $\mathcal{R}_{\mathcal{L}}(\varphi, w)$ is ZK against verifiers with b-bounded advice if for every PPT verifier V^*, there exists a PPT simulator $\mathcal{S}$ such that*

$$\{\langle P(w), V^*(z)\rangle(\varphi)\}_{\substack{(\varphi,w)\in\mathcal{R}_{\mathcal{L}} \\ z\in\{0,1\}^{b(|\varphi|)}}} \approx_c \{\mathcal{S}(z,\varphi)\}_{\substack{(\varphi,w)\in\mathcal{R}_{\mathcal{L}} \\ z\in\{0,1\}^{b(|\varphi|)}}} ,$$

where computational indistinguishability is with respect to arbitrary nonuniform distinguishers.

Remark 9 (universal simulator). In the above definition, each PPT V^* is required to have a designated PPT simulator $\mathcal{S}^*_V$. Our constructions will, in fact, guarantee the existence of one universal simulator $\mathcal{S}$ that, in addition to (z, φ), is also given the code of V^* and a bound $1^{t^*_V}$ on the running time of $V^*(\varphi; z)$, and simulates V^*'s view. Moreover, the running time of $\mathcal{S}$ is bounded by some (universal) polynomial $\mathrm{poly}(t^*_V)$ in the running time of V^*. We note that in ZK with unbounded polynomial auxiliary input, such universality follows automatically by considering the universal machine and auxiliary input $(V^*, 1^{t^*_V})$. In our context, however, this may not hold in general since t_{V^*} is unbounded and can be larger than the bound b on the size of the advice.

6.2. WIPOK with an instance-independent first message.

In this section, we define and construct 3-message WIPOKs with a particular feature: an instance-independent first message. We will later use these proof systems in our construction of a 3-message ZK argument of knowledge. In such proof systems, the prover's first message is completely independent of the statement and witness $(\varphi, w) \in \mathcal{R}_{\mathcal{L}}$ to be proven; in particular, it is of fixed polynomial size in the security parameter n, independently of the size $|\varphi, w|$ of the instance and witness.

Classical WIPOK protocols do not satisfy this requirement. For example, in the classical Hamiltonicity protocol [Blu86], the first message is independent of the witness w but does depend on the statement φ. In Lapidot and Shamir's Hamiltonicity variant [LS90], the first message is independent of (φ, w) themselves but does depend on $|\varphi, w|$ (see details in [OV12]). ZAPs, as defined by Dwork and Naor [DN07], do satisfy the independence requirement but do not constitute a proof of knowledge.

We show that relying on ZAPs and 3-message extractable commitments, we can obtain a WIPOK where the first (prover) message is completely independent of (φ, w), even of their length, and the second (verifier) message depends only on $|\varphi|$. In what follows, we formally define WIPOK with instance-independent first message, recall the definitions of ZAPs and extractable commitments, and then move to describe the construction.

DEFINITION 18 (WIPOK with instance-independent first message). *Let $\langle P, V\rangle$ be a 3-message proof system for $\mathcal{L}$, with messages denoted by $(\mathsf{wi}_1, \mathsf{wi}_2, \mathsf{wi}_3)$; we say it is a WIPOK with instance-independent first message, if it satisfies the following:*

1. Completeness with first message independence: *For any $\ell \in \mathbb{N}$, $\varphi \in \mathcal{L} \cap \{0,1\}^\ell$, $w \in \mathcal{R}_{\mathcal{L}}(\varphi)$, $n \in \mathbb{N}$,*

$$\Pr\left[V(\varphi, \mathsf{wi}_1, \mathsf{wi}_2, \mathsf{wi}_3; r') = 1 \;\middle|\; \begin{array}{r} \mathsf{wi}_1 \leftarrow P(1^n; r) \\ \mathsf{wi}_2 \leftarrow V(\ell, \mathsf{wi}_1; r') \\ \mathsf{wi}_3 \leftarrow P(\varphi, w, \mathsf{wi}_1, \mathsf{wi}_2; r) \end{array} \right] = 1 \;\; ,$$

where $r, r' \leftarrow \{0,1\}^{\mathrm{poly}(n)}$ are the randomness used by P and V.
The honest prover's first message wi_1 is of length n, independently of the length of the statement and witness (φ, w).

2. Adaptive witness-indistinguishability: *For any deterministic polynomial-size verifier V^* and all $n \in \mathbb{N}$,*

$$\Pr\left[\begin{array}{c} V^*(\varphi, \mathsf{wi}_1, \mathsf{wi}_2, \mathsf{wi}_3) = b \\ w_0, w_1 \in \mathcal{R}_{\mathcal{L}}(\varphi) \end{array} \;\middle|\; \begin{array}{r} \mathsf{wi}_1 \leftarrow P(1^n; r) \\ \varphi, w_0, w_1, \mathsf{wi}_2 \leftarrow V^*(\mathsf{wi}_1) \\ \mathsf{wi}_3 \leftarrow P(\varphi, w_b, \mathsf{wi}_1, \mathsf{wi}_2; r) \end{array} \right] \leq \frac{1}{2} + \mathrm{negl}(n) \;\; ,$$

where $b \leftarrow \{0,1\}$, $r \leftarrow \{0,1\}^{\mathrm{poly}(n)}$ is the randomness used by P.

3. Adaptive proof of knowledge: *There is a PPT extractor $\mathcal{E}$ such that for any polynomial $\ell = \ell(n)$, all $n \in \mathbb{N}$, and any deterministic prover P^*,*

$$\textit{if } \Pr\left[V(\mathsf{tr}; r') = 1 \;\middle|\; \begin{array}{r} \mathsf{wi}_1 \leftarrow P^* \\ \mathsf{wi}_2 \leftarrow V(\ell(n), \mathsf{wi}_1; r') \\ \varphi, \mathsf{wi}_3 \leftarrow P^*(\mathsf{wi}_1, \mathsf{wi}_2) \\ \mathsf{tr} = (\varphi, \mathsf{wi}_1, \mathsf{wi}_2, \mathsf{wi}_3) \end{array} \right] \geq \varepsilon \;\; ,$$

$$\textit{then } \Pr\left[\begin{array}{l} V(\mathsf{tr}; r') = 1 \\ w \leftarrow \mathcal{E}^{P^*}(1^{1/\varepsilon}, \mathsf{tr}) \\ w \notin \mathcal{R}_{\mathcal{L}}(\varphi) \end{array} \;\middle|\; \begin{array}{r} \mathsf{wi}_1 \leftarrow P^* \\ \mathsf{wi}_2 \leftarrow V(\ell(n), \mathsf{wi}_1; r') \\ \varphi, \mathsf{wi}_3 \leftarrow P^*(\mathsf{wi}_1, \mathsf{wi}_2) \\ \mathsf{tr} = (\varphi, \mathsf{wi}_1, \mathsf{wi}_2, \mathsf{wi}_3) \end{array} \right] \leq \mathrm{negl}(n) \;\; ,$$

where $\varphi \in \{0,1\}^{\ell(n)}$, and $r' \leftarrow \{0,1\}^{\mathrm{poly}(n)}$ is the randomness used by V.

DEFINITION 19 (ZAP [DN07]). *A ZAP for $\mathcal{L}$ is a public-coin, 2-message proof system $\langle P, V\rangle$, with messages denoted by (r, π), satisfying the following:*

1. Completeness: *For any $n \in \mathbb{N}$, $\varphi \in \mathcal{L} \cap \{0,1\}^n$, $w \in \mathcal{R}_{\mathcal{L}}(\varphi)$,*

$$\Pr\left[V(\varphi, \pi; r) = 1 \;\middle|\; \pi \leftarrow P(\varphi, w, r) \right] = 1 \;\; ,$$

where $r \leftarrow \{0,1\}^{\mathrm{poly}(n)}$ are the public coins of V and the probability is also over the coins of P.

2. Witness-indistinguishability: *For any deterministic polynomial-size verifier V^* and all $n \in \mathbb{N}$,*

$$\Pr\left[\begin{array}{c} V^*(\varphi,\pi)=b \\ w_0,w_1\in\mathcal{R}_{\mathcal{L}}(\varphi)\end{array}\middle|\begin{array}{r}\varphi,w_0,w_1,r\leftarrow V^* \\ \pi\leftarrow P(\varphi,w_b,r)\end{array}\right]\le\frac{1}{2}+\mathrm{negl}(n)\ ,$$

where $b\leftarrow\{0,1\}$ and the probability is also over the randomness used by P.

3. Soundness: *For any $n\in\mathbb{N}$,*

$$\Pr\left[\exists\ \begin{array}{c}\varphi\in\{0,1\}^n\setminus\mathcal{L} \\ \pi\in\{0,1\}^*\end{array}\middle| V(\varphi,\pi;r)=1\right]\le 2^{-n}\ ,$$

where the probability is over the public coins $r\leftarrow\{0,1\}^{\mathrm{poly}(n)}$.

Definition 20 (extractable commitments). *A 3-message extractable commitment scheme $(\mathcal{C},\mathcal{R})$, with messages denoted by $\vec{C}=\left(C^{(1)},C^{(2)},C^{(3)}\right)$, satisfies the following:*

1. Perfect binding: *For any transcript $\vec{C}$, there exists at most a single plaintext message m such that $\vec{C}$ is a commitment to m (relative to some random coins). The receiver $\mathcal{R}$ may abort at the end of the execution, in which case such a message may not exist.*
2. Computational hiding: *For any deterministic polynomial-size receiver $\mathcal{R}^*$,*

$$\{\langle\mathcal{C}(m_0),\mathcal{R}^*\rangle(1^n)\}_{\substack{n\in\mathbb{N}\\ m_0,m_1\in\{0,1\}^{\mathrm{poly}(n)}}}\approx_c\{\langle\mathcal{C}(m_1),\mathcal{R}^*\rangle(1^n)\}_{\substack{n\in\mathbb{N}\\ m_0,m_1\in\{0,1\}^{\mathrm{poly}(n)}}}\ ,$$

   where $\langle\mathcal{C}(m),\mathcal{R}^*\rangle(1^n)$ denotes the transcript of an execution with plaintext message m and security parameter n.
3. Extraction: *Let $\vec{C},\vec{C}'$ be any two honestly generated transcripts such that they have equal first messages $C_1=C_1'$ and distinct second messages $C_2\neq C_2'$; then it is possible to efficiently compute a plaintext message m such that $\vec{C},\vec{C}'$ are both commitments to m.*

Extractable commitments as above can be constructed from any perfectly binding noninteractive commitment (see, for example, [PW09]).

6.2.1. Construction. We now show how to use ZAPs and extractable commitments to construct a WIPOK with the required properties. As mentioned above, ZAPs already have the required independence, but they are not a proof of knowledge. The high-level idea is to add the proof of knowledge feature to ZAPs, while maintaining the required instance-independence. This can be done by having the prover commit to a random string r using a 3-message extractable commitment, and then sending, as the third message, the padded witness $w\oplus r$ along with a ZAP proof that it was computed correctly. While the first message is independent of (φ,w), it does depend on the length $|w|$; this is naturally solved by committing to a seed s of fixed length and later deriving r using a PRG.

Intuitively, extraction of the witness is now possible by extracting r (or the seed s) from the committing prover. To ensure WI, we use the idea of turning a single witness statement into a two-independent-witnesses statement as done in [FS90, COSV12, BP13]. The protocol is presented in Protocol 7 (Figure 7). In it description, we abuse notation denoting by "$\vec{C}=\mathcal{C}(m,C_2)$" the statement that $\vec{C}$ is a transcript of an execution relative to plaintext message m and receiver message C_2.

Proposition 21. *Protocol 7 is a 3-message WIPOK with instance-independent first message.*

Proof. The first message is indeed taken from a fixed distribution independently of the statement and its size. We need to show that the protocol is both WI and an argument of knowledge. The proof is an adaptation of a proof from [BP13].

Protocol 7

Common Input: security parameter n, and $\varphi \in \mathcal{L} \cap \{0,1\}^{\ell}$.

Auxiliary Input to P**:** $w \in \mathcal{R}_{\mathcal{L}}(\varphi)$.

1. P samples seeds $s_0, s_1 \leftarrow \{0,1\}^{\sqrt{n}}$, and a bit $b \leftarrow \{0,1\}$, and sends the first commitment message to each of the three $(C_0^{(1)}, C_1^{(1)}, C^{(1)}) \leftarrow (\mathcal{C}(s_0; u_0), \mathcal{C}(s_1; u_1), \mathcal{C}(b, u))$, where $|(C_0^{(1)}, C_1^{(1)}, C^{(1)})| = n$, where $(u_0, u_1, u) \leftarrow \{0,1\}^{\mathrm{poly}(n)}$ are the random coins used by the committer algorithm.[a]
2. V, given the length of the statement $\ell = |\varphi|$, samples randomness $r \leftarrow \{0,1\}^{\mathrm{poly}(\ell)}$ for a ZAP, and receiver messages $(C_0^{(2)}, C_1^{(2)}, C^{(2)}) \leftarrow (\mathcal{R}(C_0^{(1)}), \mathcal{R}(C_1^{(1)}), \mathcal{R}(C^{(1)}))$, and sends $(r, C_0^{(2)}, C_1^{(2)}, C^{(2)})$ to P.
3. P, given (φ, w):
   - computes the third committer messages $(C_0^{(3)}, C_1^{(3)}, C^{(3)}) \leftarrow (\mathcal{C}(s_0, C_0^{(2)}; u_0), \mathcal{C}(s_1, C_1^{(2)}; u_1), \mathcal{C}(b, C^{(2)}; u))$,
   - computes $a_0 = w \oplus \mathsf{PRG}(s_0), a_1 = w \oplus \mathsf{PRG}(s_1)$,
   - computes a ZAP proof π for the statement:

$$\left\{ \left\{ \vec{C} = \mathcal{C}(0, C^{(2)}) \right\} \vee \left\{ \begin{array}{c} \vec{C}_0 = \mathcal{C}(s_0, C_0^{(2)}) \\ a_0 = w_0 \oplus \mathsf{PRG}(s_0) \\ w_0 \in \mathcal{R}_{\mathcal{L}}(\varphi) \end{array} \right\} \right\} \bigwedge$$

$$\left\{ \left\{ \vec{C} = \mathcal{C}(1, C^{(2)}) \right\} \vee \left\{ \begin{array}{c} \vec{C}_1 = \mathcal{C}(s_1, C_1^{(2)}) \\ a_1 = w_1 \oplus \mathsf{PRG}(s_1) \\ w_1 \in \mathcal{R}_{\mathcal{L}}(\varphi) \end{array} \right\} \right\} \quad ;$$

     using the witness $((s_0, w), (s_1, w))$,
   - sends $C_0^{(3)}, C_1^{(3)}, C^{(3)}, a_0, a_1, \pi$.
4. V verifies the ZAP proof π, the validity of the commitments transcripts, and decides whether to accept accordingly.

[a] The commitment to b does not have to be extractable; however, we use the same commitment scheme to avoid extra notation.

FIG. 7. *A* 3*-message WIPOK with instance-independent first message*

Witness-indistinguishability. We now show that the protocol is WI. Let

$$(\bar{\varphi}, \bar{w}_0, \bar{w}_1) = \{(\varphi, w_0, w_1) : (\varphi, w_0), (\varphi, w_1) \in \mathcal{R}_{\mathcal{L}}\}$$

be any infinite sequence of instances in $\mathcal{L}$ and corresponding witness pairs. We consider a sequence of hybrids starting with a hybrid describing an interaction with a prover that uses $w_0 \in \bar{w}_0$, and ending with a hybrid describing an interaction with a prover that uses $w_1 \in \bar{w}_1$, where both w_0, w_1 are witnesses for some $\varphi \in \bar{\varphi}$. We shall prove that no efficient verifier can distinguish between any two hybrids in the sequence.

The hybrids are summarized in Table 1. We think of the hybrids as two symmetric sequences: one, 0.1-6, starts from witness w_0, and the other, 1.1-6, starts at witness w_1. We will show that within these sequences the hybrids are indistinguishable, and then we will show that 0.6 is indistinguishable from 1.6.

Hybrid 0.1. This hybrid describes a true interaction of a malicious verifier V^* with an honest prover P that uses w_0 as a witness for the statement $x \in \mathcal{L}$. In particular,

Table 1

The sequence of hybrids. The bit b corresponds to the bit commitment $\vec{C}$. The gray cells indicate the difference from the previous hybrid.

hyb	zapw_b	$\vec{C}_b$	r_b	$a_b \oplus r_b$	zapw_{1-b}	$\vec{C}_{1-b}$	r_{1-b}	$a_{1-b} \oplus r_{1-b}$
0.1	(s_b, w_0)	s_b	$\mathsf{PRG}_b(s_b)$	w_0	(s_{1-b}, w_0)	s_{1-b}	$\mathsf{PRG}(s_{1-b})$	w_0
0.2	b	s_b	$\mathsf{PRG}_b(s_b)$	w_0	(s_{1-b}, w_0)	s_{1-b}	$\mathsf{PRG}(s_{1-b})$	w_0
0.3	b	$0^{\lvert s_b \rvert}$	$\mathsf{PRG}_b(s_b)$	w_0	(s_{1-b}, w_0)	s_{1-b}	$\mathsf{PRG}(s_{1-b})$	w_0
0.4	b	$0^{\lvert s_b \rvert}$	u	w_0	(s_{1-b}, w_0)	s_{1-b}	$\mathsf{PRG}(s_{1-b})$	w_0
0.5	b	$0^{\lvert s_b \rvert}$	u	w_1	(s_{1-b}, w_0)	s_{1-b}	$\mathsf{PRG}(s_{1-b})$	w_0
0.6	(s_b, w_1)	s_b	$\mathsf{PRG}_b(s_b)$	w_1	(s_{1-b}, w_0)	s_{1-b}	$\mathsf{PRG}(s_{1-b})$	w_0
1.6	(s_b, w_0)	s_b	$\mathsf{PRG}_b(s_b)$	w_0	(s_{1-b}, w_1)	s_{1-b}	$\mathsf{PRG}(s_{1-b})$	w_1
1.2-5	...	...	...	...	...	...	...	...
1.1	(s_b, w_1)	s_b	$\mathsf{PRG}_b(s_b)$	w_1	(s_{1-b}, w_1)	s_{1-b}	$\mathsf{PRG}(s_{1-b})$	w_1

the ZAP proof is given using the witness $((s_0, w_0), (s_1, w_0))$.[8] In Table 1, the witness used in part 0 of the ZAP is referred to as zapw_0, and the one corresponding to 1 is zapw_1.

Hybrid 0.2. This hybrid differs from the previous one only in the witness used in the ZAP. Specifically, for the bit b given by $\vec{C}$, the witness for the ZAP is set to be $(b, (s_{1-b}, w_0))$, instead of $((s_b, w_0), (s_{1-b}, w_0))$.[9] Since the ZAP is WI, this hybrid is computationally indistinguishable from the previous one.

Hybrid 0.3. In this hybrid, the commitment $\vec{C}_b$ is for the plaintext $0^{|s_b|}$, instead of the plaintext s_b. This hybrid is computationally indistinguishable from the previous one due to the computational hiding of the commitment scheme $\vec{C}$.

Hybrid 0.4. In this hybrid, instead of padding with $\mathsf{PRG}(s_b)$, padding is done with a random independent string $r_b \leftarrow \{0,1\}^{|\mathsf{PRG}(s_b)|}$. Computational indistinguishability of this hybrid and the previous one follows from the pseudorandomness of the PRG.

Hybrid 0.5. In this hybrid, the padded value a_b is taken to be $w_1 \oplus r_b$, instead of $w_0 \oplus r_b$. Since r_b is now uniform and independent of all other elements, this hybrid induces the exact same distribution as the previous hybrid.

Hybrid 0.6. This hybrid now backtracks, returning to the same experiment as in Hybrid 0.1 with the exception that the ZAP witness is now $((s_b, w_1), (s_{1-b}, w_0))$ instead of $((s_b, w_0), (s_{1-b}, w_0))$. This indistinguishability follows exactly as when moving from 0.1 to 0.5 (only backward).

Hybrids 1.1 *to* 1.6. These hybrids are symmetric to the above hybrids, only they start from w_1 instead of w_0. This means that they end in 1.6, which uses a ZAP witness $((s_b, w_0), (s_{1-b}, w_1))$, which is the same as 0.6, only in reverse order.

Hybrids 0.6 *and* 1.6 *are computationally indistinguishable.* This follows directly from the computational hiding of the commitment $\vec{C}$ to b. Indeed, assume toward contradiction that V distinguishes the two hybrids. Concretely, denote the probability it outputs 1 on 0.6 by p_0 and the probability it outputs 1 on 1.6 by p_1, and assume w.l.o.g. that $p_0 - p_1 \geq \varepsilon(n)$ for some noticeable $\varepsilon(n)$. We can construct a predictor that given a commitment $\vec{C} = \mathcal{C}(b)$ to a random bit $b \leftarrow \{0,1\}$ guesses b with probability $\frac{1+\varepsilon(n)}{2}$. The predictor samples a random $b' \leftarrow \{0,1\}$ as a candidate guess for b, and performs the experiment corresponding to 0.6 only it locates w_0 and w_1 according to

[8] Formally, the witness also includes the randomness for the commitments $\vec{C}_0$ and $\vec{C}_1$, but for notational brevity, we shall omit it.

[9] Again the witness should include the randomness for the commitment $\vec{C}$, and $\vec{C}_{1-b}$, but this is omitted from our notation.

b', rather than the unknown b. If the distinguisher outputs 1, the predictor guesses $b = b'$, and otherwise it guesses $b = 1 - b'$.

Conditioned on $b = b'$, V is experiencing 0.6, and thus the guess will be correct with probability p_0; conditioned on $b = 1 - b'$, V is experiencing 1.6, and the guess will be right with probability $1 - p_1$. So overall the guessing probability is $\frac{p_0}{2} + \frac{1-p_1}{2} \geq \frac{1}{2} + \frac{\varepsilon(n)}{2}$.

This completes the proof that the protocol is WI.

Argument of knowledge. We show that the protocol is an argument of knowledge. Indeed, let P^* be any prover that convinces the honest verifier of accepting with noticeable probability $\varepsilon(n)$; then with probability at least $\varepsilon(n)/2$ over its first message, it holds with probability at least $\varepsilon(n)/2$ over the rest of the protocol that P^* convinces V. Let us call such a prefix (namely, first message) good. Now, for any good prefix, we can consider the perfectly binding induced commitment to the bit b, and from the soundness of the ZAP, we get a circuit that with probability at least $\varepsilon(n)/2 - \mathrm{negl}(n)$ produces an accepting commitment transcript for the plaintext s_{1-b} and gives a valid witness $w \in \mathcal{R}_{\mathcal{L}}$, padded with $\mathsf{PRG}(s_{1-b})$. This in particular means that we can first sample a prefix (hope it is good), and then use the extraction guarantee of the commitment to learn s_{1-b} and $\mathsf{PRG}(s_{1-b})$, and thus also the witness w.

This completes the proof of Proposition 21. □

2-*message WI with instance-independent first message.* We shall also make use of 2-message WI with instance-independent first message. Here, there are two verifier and prover messages. Like in the 3-message definition the verifier message does not depend on the instance but is allowed to depend on its length. In such a protocol, we require only soundness. ZAPs, for instance, satisfy this requirement, but we can also do with a privately verifiable protocol rather than a ZAP. (In fact, also in the above construction of 3-message WIPOKs with instance-independent first message, the ZAPs can be replaced with any 2-message WI with an instance-independent first message.)

6.3. 1-hop homomorphic Encryption.

A 1-*hop homomorphic encryption scheme* [GHV10] allows a pair of parties to securely evaluate a function as follows: the first party encrypts an input, the second party homomorphically evaluates a function on the ciphertext, and the first party decrypts the evaluation result. Such a scheme can be instantiated based on garbled circuits and an appropriate 2-message oblivious transfer protocol, based on either decision Diffie–Hellman or quadratic residuosity [Yao86, GHV10, NP01, AIR01, HK12].

DEFINITION 22. $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Eval}, \mathsf{Dec})$, *where* $\mathsf{Gen}, \mathsf{Eval}$ *are probabilistic and* Enc, Dec *are deterministic, is a semantically secure, circuit-private,* 1-*hop homomorphic encryption scheme if it satisfies the following properties:*

- Perfect correctness: *For any $n \in \mathbb{N}$, $x \in \{0,1\}^n$ and circuit C,*

$$\Pr_{\substack{\mathsf{sk}\leftarrow\mathsf{Gen}(1^n)\\ \mathsf{c}=\mathsf{Enc}_{\mathsf{sk}}(x)\\ \mathsf{Eval}}}\left[\begin{array}{c}\hat{\mathsf{c}} \leftarrow \mathsf{Eval}(\mathsf{c}, C)\\ \mathsf{Dec}_{\mathsf{sk}}(\hat{\mathsf{c}}) = C(x)\end{array}\right] = 1 \ .$$

- Semantic security: *For any polynomial-size $\mathcal{A}$, $n \in \mathbb{N}$, and any pair of inputs $x_0, x_1 \in \{0,1\}^n$*

$$\Pr_{\substack{\mathsf{b}\leftarrow\{0,1\}\\ \mathsf{sk}\leftarrow\mathsf{Gen}(1^n)}}\left[\mathcal{A}(\mathsf{Enc}_{\mathsf{sk}}(x_{\mathsf{b}})) = \mathsf{b}\right] < \frac{1}{2} + \mathrm{negl}(n) \ .$$

- Circuit privacy: *A randomized evaluation should not leak information on the input circuit C. This should hold even for malformed ciphertexts. Formally, let $\mathcal{E}(x) = \mathsf{Supp}(\mathsf{Enc}(x))$ be the set of all legal encryptions of x, let $\mathcal{E}_n = \cup_{x\in\{0,1\}^n}\mathcal{E}(x)$ be the set legal encryptions for strings of length n, and let $\mathcal{C}_n$ be the set of all circuits on n input bits. There exists a (possibly unbounded) simulator $\mathcal{S}_{\mathsf{1hop}}$ such that*

$$\{C, \mathsf{Eval}(c, C)\}_{\substack{n\in\mathbb{N}, C\in\mathcal{C}_n\\ x\in\{0,1\}^n, c\in\mathcal{E}(x)}} \approx_c \{C, \mathcal{S}_{\mathsf{1hop}}(c, C(x), |C|)\}_{\substack{n\in\mathbb{N}, C\in\mathcal{C}_n,\\ x\in\{0,1\}^n, c\in\mathcal{E}(x)}},$$

$$\{C, \mathsf{Eval}(c, C)\}_{\substack{n\in\mathbb{N}\\ C\in\mathcal{C}_n, c\notin\mathcal{E}_n}} \approx_c \{C, \mathcal{S}_{\mathsf{1hop}}(c, \perp, |C|)\}_{\substack{n\in\mathbb{N}\\ C\in\mathcal{C}_n, c\notin\mathcal{E}_n}}.$$

6.4. Constructions. In this section, we construct ZK protocols against verifiers with bounded advice from generalized EOWFs against adversaries with bounded auxiliary input (GEOWFs against bounded auxiliary-input adversaries). We start by describing a construction of a 3-message argument of knowledge from any GEOWF, 1-hop homomorphic encryption, and 3-message WIPOK with instance-independent first message. We then show a 2-message argument, assuming (noninteractive) commitments that can be inverted in super-poly time $T(n)$, GEOWFs that are hard against poly$(T(n))$-size adversaries, and any 2-message WI with instance-independent verifier message (in particular, ZAPs).

6.4.1. A 3-message ZK argument of knowledge.
Ingredients and notation.

- A semantically secure, circuit-private, 1-hop homomorphic encryption scheme $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Eval}, \mathsf{Dec})$.
- A 3-message WIPOK with an instance-independent first message with messages denoted by $(\mathsf{wi}_1, \mathsf{wi}_2, \mathsf{wi}_3)$.
- A keyless GEOWF $\mathcal{F} = \{f_n\}_{n\in\mathbb{N}}$, against $(b(n)+2n)$-bounded-auxiliary-input adversaries, with respect to a privately verifiable relation $\mathcal{R}^{\mathcal{F}} = \{\mathcal{R}_n^{\mathcal{F}}\}_{n\in\mathbb{N}}$. We denote by $\mathcal{T}$ the polynomial-time tester such that $\mathcal{T}(x, x') = \mathcal{R}_n^{\mathcal{F}}(f_n(x), x')$.
- We denote by $\mathcal{T}_{y,x'}(x)$ a circuit that, given input x, verifies that "$y \neq f_n(x)$" or "$\mathcal{T}(x, x') = 1$"; that is, either "x is not a valid preimage of y" or "$\mathcal{R}_n^{\mathcal{F}}(f_n(x), x') = 1$."
- We denote by $\mathbf{1} = \mathbf{1}_{y,x'}$ a circuit of the same size as $\mathcal{T}_{y,x'}$ that always returns 1.

The protocol is given in Figure 8.

THEOREM 23. *Protocol* 8 *is a ZK argument of knowledge against b-bounded-auxiliary-input verifiers.*

High-level idea behind the proof. For simplicity let us explain why the protocol is sound; showing it is an argument of knowledge follows a similar reasoning. Assuming that $\varphi \notin \mathcal{L}$, in order to pass the WIPOK, with respect to an evaluated cipher $\hat{\mathsf{c}}$ that decrypts to 1, the prover must know a witness x' such that $\mathcal{T}_{y,x'}(x) = 1$. This, by definition, and the fact that the verifier indeed sends an image $y = f_n(x)$ together with its encrypted preimage x, means that x' must satisfy the relation $\mathcal{R}^{\mathcal{F}}(f_n(x), x') = 1$, and thus the prover violates $\mathcal{R}^{\mathcal{F}}$-hardness. (Formally, we also need to invoke semantic security to claim that the encryption of x does not help in producing such a witness.)

To show ZK, we rely on the fact that if the verifier sends y together with an encryption of a true preimage x, the simulator can invoke the extractor and extract a witness x' from its code and auxiliary input and use it to complete the WIPOK. Here we use the bound on the first WI prover message to claim that the overall

Protocol 8

Common Input: $\varphi \in \mathcal{L} \cap \{0,1\}^n$.

Auxiliary Input to P: a witness w for φ.

1. P computes
   - the first message $\mathsf{wi}_1 \in \{0,1\}^n$ of the instance-dependent WIPOK,

   and sends wi_1.
2. V computes
   - a random preimage $x \leftarrow \{0,1\}^{\ell(n)}$,
   - a corresponding image $y = f_n(x)$,
   - a secret key for a 1-hop homomorphic encryption $\mathsf{sk} \leftarrow \mathsf{Gen}(1^n)$,
   - an encryption $\mathsf{c}_x = \mathsf{Enc}_{\mathsf{sk}}(x)$,
   - the second WIPOK message wi_2 for statements of size ℓ_Ψ, where ℓ_Ψ is determined below,

   and sends $(y, \mathsf{c}_x, \mathsf{wi}_2)$.
3. P computes
   - a homomorphic evaluation $\hat{\mathsf{c}} \leftarrow \mathsf{Eval}(\mathbf{1}, \mathsf{c}_x)$,
   - the third WIPOK message wi_3 for statement $\Psi = \Psi_1(\varphi) \vee \Psi_2(y, \mathsf{c}_x, \hat{\mathsf{c}})$ stating that:

     $$\{\varphi \in \mathcal{L}\} \bigvee \left\{\exists (x', r) : \hat{\mathsf{c}} = \mathsf{Eval}(\mathcal{T}_{y,x'}, \mathsf{c}_x; r)\right\} \quad .$$

     The proof is computed using the witness $w \in \mathcal{R}_{\mathcal{L}}(\varphi)$.

   It sends $(\hat{\mathsf{c}}, \mathsf{wi}_3)$.
4. V verifies the WIPOK and that $\mathsf{Dec}_{\mathsf{sk}}(\hat{\mathsf{c}}) = 1$.

FIG. 8. *A 3-message ZK argument of knowledge against verifiers with b-bounded auxiliary-input.*

auxiliary input is appropriately bounded. In case that the verifier diverges from the protocol, and does not send a proper y and encrypted preimage, the definition of $\mathcal{T}_{y,x'}$ guarantees that the circuit will also accept. Thus, in either case, the circuit privacy of homomorphic evaluation would guarantee indistinguishability from a real proof, where the prover actually evaluates the constant $\mathbf{1}$ circuit.

A more detailed proof follows.

Proof. We first show that the protocol is an argument of knowledge.

PROPOSITION 24. *Protocol* 8 *is an argument of knowledge arbitrary polynomial-size provers.*

Proof. We show that there exists a PPT extractor $\mathcal{E}$ such that for any polynomial-size prover P^* that generates $\varphi_n \in \{0,1\}^n$ and convinces V of accepting φ_n with nonnegligible probability $\varepsilon(n) = n^{-O(1)}$, the extractor $\mathcal{E}^{P^*}(1^{\varepsilon(n)}, \varphi_n)$ outputs $w \in \mathcal{R}_{\mathcal{L}}(\varphi_n)$ with probability $\varepsilon(n)^2/4 - \mathrm{negl}(n)$.[10] We start by describing the extractor and then analyze its performance. Throughout the description (and following proof), we will often omit n, when it is clear from the context.

The witness extractor $\mathcal{E}^{P^*}(1^{\varepsilon(n)}, \varphi_n)$ operates as follows:

1. Derives from P^* a new prover P^*_{wi} for the WIPOK as follows. P^*_{wi} emulates the role of P^* in the WIPOK; in particular, it would (honestly) sample (y, c_x) on its own to compute the second verifier message $(y, \mathsf{c}_x, \mathsf{wi}_2,)$ that P^* receives.
2. Chooses the random coins r for P^*_{wi} and samples a transcript $\mathsf{tr} = (\Psi, \mathsf{wi}_1, \mathsf{wi}_2, \mathsf{wi}_3)$ of an execution with the honest WIPOK verifier V_{wi}.

[10] The extraction probability can then be amplified to $1 - \mathrm{negl}(n)$ by standard repetition.

3. Applies the WIPOK extractor $\mathcal{E}_{\mathsf{wi}}$ on the transcript tr, with oracle access to P^*_{wi}, and extraction parameter $2/\varepsilon$. That is, computes $w \leftarrow \mathcal{E}_{\mathsf{wi}}^{P^*_{\mathsf{wi}}(r)}(1^{2/\varepsilon}, \mathsf{tr})$.
4. Outputs w.

We will show that if the extractor fails to extract with the required probability, then we can use P^* to break the $\mathcal{R}^{\mathcal{F}}$-hardness of $\mathcal{F}$. Thus from here on, we assume that for some noticeable function $\eta(n) = n^{-O(1)}$, with probability at most $\varepsilon^2/4 - \eta$ the extracted witness w is in $\mathcal{R}_{\mathcal{L}}(\varphi)$. We shall first establish several claims regarding the extraction procedure and the consequences of extraction failure. Then we will describe the reduction to $\mathcal{R}^{\mathcal{F}}$-hardness.

We start by noting that an execution of $P^*_{\mathsf{wi}}(r)$ with the honest WIPOK verifier V_{wi} induces a perfectly emulated execution of P^* with the honest verifier V. Thus, we know that V, and in particular V_{wi}, accept in such an execution with probability $\varepsilon(n) \geq n^{-O(1)}$.

Good coins r. We say that random coins r for P^*_{wi} are good if with probability at least $\varepsilon/2$ over the coins of the WIPOK verifier V_{wi}, the induced execution of P^* with V is such that the ZK verifier V accepts. By a standard averaging argument, at least an $(\varepsilon/2)$-fraction of the coins r for P^*_{wi} are good.

Recall that every execution of $\mathcal{E}_{\mathsf{wi}}$ induces a choice r for P^*_{wi}, a WIPOK transcript $\mathsf{tr} = (\Psi, \mathsf{wi}_1, \mathsf{wi}_2, \mathsf{wi}_3)$, and values $(y, \mathsf{c}_x, \hat{\mathsf{c}})$ exchanged in the induced interaction between the ZK prover P^* and the ZK verifier V. These values, in turn, determine the statement

$$\Psi = \Psi_1(\varphi) \vee \Psi_2(y, \mathsf{c}_x, \hat{\mathsf{c}}).$$

We next claim that for any good r, such an extraction procedure outputs a witness for Ψ and simultaneously the homomorphic evaluation result $\hat{\mathsf{c}}$ decrypts to one (under the secret key sk sampled together with c_x), with nonnegligible probability.

Claim 3 (extraction for good r). *For any good r for P^*_{wi}, it holds that w satisfies the induced statement Ψ and $\mathsf{Dec}_{\mathsf{sk}}(\hat{\mathsf{c}}) = 1$ with probability $\varepsilon(n)/2 - \operatorname{negl}(n)$ over a transcript tr, and coins for $\mathcal{E}_{\mathsf{wi}}$.*

Proof of Claim 3. Fix some good coins r. Since the coins r are good, the WIPOK verifier V_{wi} is convinced by P^*_{wi} with probability at least $\varepsilon/2$, meaning that V_{wi} accepts and in addition $\mathsf{Dec}_{\mathsf{sk}}(\hat{\mathsf{c}}) = 1$. We claim that when this occurs then, except with probability $\operatorname{negl}(n)$, the extractor $\mathcal{E}_{\mathsf{wi}}$, also outputs a valid witness w for Ψ. This follows directly from the extraction guarantee of the WIPOK. □

Now, relying on the fact that overall the extractor fails to output a witness for φ, we deduce that with nonnegligible probability, the extracted witness satisfies the trapdoor statement Ψ_2.

Claim 4 (extracting a trapdoor witness). *In a random execution of the extractor, the extracted witness w satisfies the trapdoor statement $\Psi_2(y, \mathsf{c}_x, \hat{\mathsf{c}})$, and in addition $\mathsf{Dec}_{\mathsf{sk}}(\hat{\mathsf{c}}) = 1$, with probability at least $\eta(n) - \operatorname{negl}(n)$ over the choice of r for P^*_{wi}, a transcript tr, and coins for $\mathcal{E}_{\mathsf{wi}}$.*

Proof of Claim 4. First, by the $(\varepsilon/2)$-density of good r's and Claim 3, we deduce that in a random execution the extracted w satisfies the statement $\Psi = \Psi_1 \vee \Psi_2$, and in addition $\mathsf{Dec}_{\mathsf{sk}}(\hat{\mathsf{c}}) = 1$, with probability at least $\varepsilon^2/4 - \operatorname{negl}(n)$. Combining this with the fact that $w \in \mathcal{R}_{\mathcal{L}}(\varphi)$ with probability at most $\varepsilon^2/4 - \eta$, the claim follows. □

Next, recall that by the definition of Ψ_2, whenever w is a witness for Ψ_2, it holds that

$$w = (x', r) \text{ such that } \hat{\mathsf{c}} = \mathsf{Eval}(\mathcal{T}_{y,x'}, \mathsf{c}_x; r) \ .$$

Furthermore, by the definition of $\mathcal{T}_{y,x'}$ and the perfect correctness of the 1-hop homomorphic encryption,

$$\mathsf{Dec}_{\mathsf{sk}}(\hat{\mathsf{c}}) = \mathcal{T}_{y,x'}(x) = \mathcal{T}(x, x') \ .$$

We thus deduce that, with probability η, the witness $w = (x', r)$ extracted by $\mathcal{E}$ is such that $\mathcal{R}_n^{\mathcal{F}}(y, x') = 1$.

An equivalent experiment that hides the secret input x. We now consider an augmented extraction procedure $\mathcal{E}_{\mathsf{aug}}$ that behaves exactly as the original extractor $\mathcal{E}$, except that, when P^*_{wi} emulates P^*, it does not sample an encryption c_x of the secret verification state x, but rather it samples an encryption c_0 of $0^{|x|}$. We claim that in this alternative experiment, the above condition still holds with the same probability up to a negligible difference.

CLAIM 5 (success probability in alternative experiment.). *With probability $\eta - \operatorname{negl}(n)$, the witness $w = (x', r)$ extracted by $\mathcal{E}_{\mathsf{aug}}$ is such that $\mathcal{R}_n^{\mathcal{F}}(y, x') = 1$.*

Proof sketch of Claim 5. This claim follows from the semantic security of the 1-hop homomorphic encryption scheme. Indeed, if the above was not the case, we can distinguish between an encryption of x and one of $0^{|x|}$. For this, note that the first experiment with c_x (respectively, the second with c_0) can be perfectly emulated given x and the ciphertext c_x (respectively, c_0), and in addition the above condition can be tested efficiently. □

The reduction to $\mathcal{R}^{\mathcal{F}}$-hardness. We are now ready to describe a reduction $\mathcal{R}$ that violates $\mathcal{R}^{\mathcal{F}}$-hardness. For this, we consider a restricted augmented extractor $\mathcal{E}_{\mathsf{aug}}^{y^*}$ that unlike $\mathcal{E}_{\mathsf{aug}}$ does not emulate the image $y = f_n(x)$ itself but uses instead y^*. The reduction $\mathcal{R}^{P^*}$, given a challenge $y^* = f_n(x^*)$, invokes $\mathcal{E}_{\mathsf{aug}}^{y^*, P^*}(1^{\varepsilon(n)}, \varphi_n)$, obtains a witness $w = (x', r)$, and outputs x'.

CLAIM 6. *$\mathcal{R}$ finds x' such that $\mathcal{R}_n^{\mathcal{F}}(y^*, x') = 1$ with probability $\eta - \operatorname{negl}(n)$.*

Proof. For a random challenge $y^* = f_n(x^*)$, the experiment in which $\mathcal{E}_{\mathsf{aug}}^{y^*, P^*}(1^{\varepsilon(n)}, \varphi_n)$ is invoked is distributed identically to that where $\mathcal{E}_{\mathsf{aug}}^{P^*}(1^{\varepsilon(n)}, \varphi_n)$ (which samples y on its own) is invoked. Thus, the claim follows from Claim 5.

This completes the proof of Proposition 24. □

We next show that the protocol is ZK.

PROPOSITION 25. *Protocol* 8 *is ZK against any polynomial-time verifier V^* with auxiliary input of size at most $b(n)$.*

Bounded randomness. We note that, since the ZK simulator is allowed to simulate the (a priori unbounded) randomness of the verifier V^*, we can restrict attention to verifiers V^* that only have bounded randomness. Indeed, assuming the existence of one-way functions, we can always consider a new verifier $\widetilde{V}^*$ that first stretches its bounded randomness using a PRG and then emulates V^* using the generated pseudorandom string. To simulate the view of V^*, we can first apply the simulator $\widetilde{\mathcal{S}}$ for $\widetilde{V}^*$ and then apply the PRG on the simulated randomness to obtain a full simulated view for V^*. Accordingly, from here on we can simply focus on deterministic verifiers V^* that get their bounded randomness as part of their bounded advice.

Proof of Proposition 25. We describe a universal (Remark 9) ZK simulator $\mathcal{S}$ and show its validity. Let $\varphi \in \mathcal{L}$ and let V^* be the code of any malicious verifier with running time bounded by $t(n)$, and let z' be any advice of length at most $b(n)$.

The simulator $\mathcal{S}(V^*, 1^{t(n)}, \varphi, z')$, where $|\varphi| = n$, operates as follows:

1. Generates the first message $\mathsf{wi}_1 \in \{0,1\}^n$ of the WIPOK.
2. Feeds wi_1 to $V^*(\varphi; z')$, who returns $(y, \mathsf{c}, \mathsf{wi}_2)$ that are (allegedly) an image under the function f_n, an encryption of a corresponding preimage, and the second message of the WIPOK.
3. Computes the third message $(\hat{\mathsf{c}}, \mathsf{wi}_3)$ as follows:
   (a) Constructs from the code of V^* a machine $\mathcal{M}_{V^*}$ that, given 1^n and $z = (z', \varphi, \mathsf{wi}_1)$ as input, outputs some y, and whose running time is linear in the running time t_{V^*} of V^*.
   (b) Applies the extractor $\mathcal{E}$ on $\mathcal{M}_{V^*}$ to obtain a witness x' such that $\mathcal{R}^{\mathcal{F}}(y, x') = 1$.
   (c) Computes $\hat{\mathsf{c}} = \mathsf{Eval}(\mathcal{T}_{y,x'}, \mathsf{c}; r)$ using randomness r.
   (d) Computes the third WIPOK message wi_3 for the statement $\Psi = \Psi_1(\varphi) \vee \Psi_2(y, \mathsf{c}_x, \hat{\mathsf{c}})$ given by
   $$\{\varphi \in \mathcal{L}\} \bigvee \{\exists (x', r) : \hat{\mathsf{c}} = \mathsf{Eval}(\mathcal{T}_{y,x'}, \mathsf{c}_x; r)\}$$
   using the witness (x', r) for the trapdoor statement Ψ_2.
   (e) Outputs the view $(\mathsf{wi}_1, \hat{\mathsf{c}}, \mathsf{wi}_3)$ of V^*.

Note that $|z| \leq |z'| + |\varphi| + |\mathsf{wi}_1| \leq b(n) + 2n$, and thus, if $y = f_n(x)$ for some x, applying the extractor $\mathcal{E}$ on $\mathcal{M}_{V^*}$ would result in a witness x', such that $\mathcal{R}^{\mathcal{F}}(y, x') = 1$, in time poly$(t_V^*)$. ($\mathcal{S}$ does not test whether y is a valid image; it applies the extractor regardless to obtain a candidate x'.) We now show that the view generated by $\mathcal{S}$ is computationally indistinguishable from the view of V^* in an execution with the honest prover P. We do this by exhibiting a sequence of hybrids.

Hybrid 1. The view $(\mathsf{wi}_1, \hat{\mathsf{c}}, \mathsf{wi}_3)$ is generated by $\mathcal{S}$.

Hybrid 2. Instead of generating wi_3, using the witness (x', r) for Ψ_2, it is generated using a witness w for $\Psi_1 = \{\varphi \in \mathcal{L}\}$. By the adaptive witness-indistinguishability of the WIPOK system, this hybrid is computationally indistinguishable from Hybrid 1.

Hybrid 3. The view $(\mathsf{wi}_1, \hat{\mathsf{c}}, \mathsf{wi}_3)$ is generated in an interaction of V^* with the honest prover P. The difference from Hybrid 3 is that $\hat{\mathsf{c}}$ is sampled from $\mathsf{Eval}(\mathbf{1}, \mathsf{c}_x)$ instead of $\mathsf{Eval}(\mathcal{T}_{y,x'}, \mathsf{c}_x)$ for the extracted input x'.

First, in case c is a well-formed ciphertext and $x = \mathsf{Dec}_{\mathsf{sk}}(\mathsf{c})$, the circuit privacy of the 1-hop homomorphic encryption guarantees that

$$\mathsf{Eval}(\mathcal{T}_{y,x'}, \mathsf{c}_x) \approx_c \mathcal{S}_{\mathsf{1hop}}(\mathsf{c}_x, \mathcal{T}_{y,x'}(x), |\mathcal{T}_{y,x'}|) \equiv \mathcal{S}_{\mathsf{1hop}}(\mathsf{c}_x, \mathbf{1}(x), |\mathbf{1}|) \approx_c \mathsf{Eval}(\mathbf{1}, \mathsf{c}_x) \ .$$

To see that indeed $\mathcal{T}_{y,x'}(x) = \mathbf{1}(x)$, note that if $y = f_n(x)$ the extracted x' is such that $\mathcal{T}(x, x') = 1$. From the definition of $\mathcal{T}_{y,x'}$, it follows that $\mathcal{T}_{y,x'}(x) = 1$.

Also, for any malformed ciphertext c^* it holds that

$$\mathsf{Eval}(\mathcal{T}_{y,x'}, \mathsf{c}^*) \approx_c \mathcal{S}_{\mathsf{1hop}}(\mathsf{c}^*, \perp, |\mathcal{T}_{y,x'}|) \equiv \mathcal{S}_{\mathsf{1hop}}(\mathsf{c}^*, \perp, |\mathbf{1}|) \approx_c \mathsf{Eval}(\mathbf{1}, \mathsf{c}^*).$$

It follows that Hybrid 3 is computationally indistinguishable from Hybrid 2. □

This completes the proof of Theorem 23. □

6.4.2. A 2-message ZK argument. In this section, we show that, using complexity leveraging (and superpolynomial hardness assumptions), we can augment the protocol from the previous section to a 2-message argument.

Protocol 9

Common Input: $\varphi \in \mathcal{L} \cap \{0,1\}^n$.

Auxiliary Input to P: a witness w for φ.

1. V computes
    - a random preimage $x \leftarrow \{0,1\}^{\ell(n)}$,
    - a corresponding image $y = f_n(x)$,
    - a secret key for a 1-hop homomorphic encryption $\mathsf{sk} \leftarrow \mathsf{Gen}(1^n)$,
    - an encryption $\mathsf{c}_x = \mathsf{Enc}_{\mathsf{sk}}(x)$,
    - the first WI message wi_1 for statements of size ℓ_Ψ, where ℓ_Ψ is determined below,

    and sends $(y, \mathsf{c}_x, \mathsf{wi}_1)$.
2. P computes
    - a commitment to zero $C \leftarrow \mathcal{C}(0^\ell)$,
    - a homomorphic evaluation $\hat{\mathsf{c}} \leftarrow \mathsf{Eval}(\mathbf{1}, \mathsf{c}_x)$,
    - the second WI message wi_2 for statement $\Psi = \Psi_1(\varphi) \vee \Psi_2(y, \mathsf{c}_x, \hat{\mathsf{c}}, C)$ stating that:

    $$\{\varphi \in \mathcal{L}\} \bigvee \left\{ \exists (x', r, r') : \begin{array}{c} \hat{\mathsf{c}} = \mathsf{Eval}(\mathcal{T}_{y,x'}, \mathsf{c}_x; r) \\ C = \mathcal{C}(x'; r') \end{array} \right\} .$$

    The proof is computed using the witness $w \in \mathcal{R}_{\mathcal{L}}(\varphi)$.

    It sends $(C, \hat{\mathsf{c}}, \mathsf{wi}_2)$.
3. V verifies the WI and that $\mathsf{Dec}_{\mathsf{sk}}(\hat{\mathsf{c}}) = 1$.

FIG. 9. *A 2-message ZK argument against verifiers with b-bounded auxiliary input.*

Ingredients and notation.

- A semantically secure, circuit-private, 1-hop homomorphic encryption scheme $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Eval}, \mathsf{Dec})$.
- A 2-message WI with an instance-independent first message with messages denoted by $(\mathsf{wi}_1, \mathsf{wi}_2)$.
- A keyless GEOWF $\mathcal{F} = \{f_n\}_{n\in\mathbb{N}}$, against $(b(n)+n)$-bounded-auxiliary-input adversaries, with respect to a privately verifiable relation $\mathcal{R}^{\mathcal{F}} = \left\{\mathcal{R}^{\mathcal{F}}_n\right\}_{n\in\mathbb{N}}$. We denote by $\mathcal{T}$ the polynomial-time tester such that $\mathcal{T}(x,x') = \mathcal{R}^{\mathcal{F}}_n(f_n(x), x')$. Further assume that $\mathcal{F}$ is one-way against adversaries of size $\mathrm{poly}(T)$ (see Remark 8).
- A perfectly binding commitment $\mathcal{C}$ that is hiding against polynomial-size adversaries, and can be completely inverted in time $T(n)$, for some computable superpolynomial function $T(n) = n^{\omega(1)}$.
- We denote by $\mathcal{T}_{y,x'}(x)$ a circuit that, given input x, verifies that "$y \neq f_n(x)$" or "$\mathcal{T}(x,x') = 1$"; that is, either "x is not a valid preimage of y" or $\mathcal{R}^{\mathcal{F}}_n(f_n(x), x') = 1$.
- We denote by $\mathbf{1} = \mathbf{1}_{y,x'}$ a circuit of the same size as $\mathcal{T}_{y,x'}$ that always returns 1.

The protocol is given in Figure 9.

THEOREM 26. *Protocol* 9 *is a ZK argument against b-bounded-auxiliary-input verifiers.*

High-level idea behind the proof. Proving ZK against verifiers with bounded advice is essentially the same as in the 3-message protocol, only now the simulator also commits to the input that it extracts from the verifier (by the hiding of the commitment ZK is maintained). The proof of soundness is essentially the same as showing proof of knowledge in the 3-message protocol, only now the WI proof does not provide witness extraction; instead we will extract a witness in time $\mathrm{poly}(T(n))$ by inverting the prover's commitment with brute-force. Since one-wayness holds even against $\mathrm{poly}(T(n))$-adversaries, soundness follows.

The actual proof closely follows the proof of Theorem 23. We now give a proof sketch, stressing the differences between the two.

Proof sketch. We first show that the protocol is sound against polynomial-size adversaries.

CLAIM 7. *Protocal* 9 *is an argument.*

Proof sketch. Let P^* be any polynomial-size prover, and assume toward contradiction that for infinitely many $\varphi \notin \mathcal{L}$, P^* convinces V of accepting with noticeable probability $\varepsilon(n)$. We show that P^* can be used to break the $\mathcal{R}^{\mathcal{F}}$-hardness of $\mathcal{F}$. The breaker, given the image y, would sample a first WI message wi_1, and an encryption of zero c_0, and feed $(y, \mathsf{c}_0, \mathsf{wi}_1)$ to P^*, who would then output a commitment C, an (alleged) image y, and a proof wi_2 for the statement $\Psi = \Psi_1(\varphi) \vee \Psi_2(y, \mathsf{c}_0, \hat{\mathsf{c}}, C)$. The breaker would now invert C in time $T(n)$ and output the result x'.

We now argue that, with probability $\varepsilon(n) - \mathrm{negl}(n)$, C is a commitment to a witness x' satisfying $\mathcal{R}^{\mathcal{F}}_n(f_n(x), x') = 1$, thus breaking the $\mathcal{R}^{\mathcal{F}}$-hardness of $\mathcal{F}$, against $\mathrm{poly}(T(n))$-time adversaries. To see this, note that by the semantic security of the 1-hop encryption, the described experiment is indistinguishable from an experiment in which the breaker uses an encryption c_x of an actual preimage x of y, as in a real interaction between V and P^*. Thus, as in a real interaction, with probability $\varepsilon(n) - \mathrm{negl}(n)$, the WI proof computed by the breaker is convincing. Since $\varphi \notin \mathcal{L}$, it follows by the soundness of the WI system the statement $\Psi_2(y, \mathsf{c}_0, \hat{\mathsf{c}}, C)$ holds, and thus in particular, C is a commitment to a witness x' such that $\mathcal{R}^{\mathcal{F}}_n(f_n(x), x') = \mathcal{T}_{y,x'} = 1$. □

We next show that the protocol is ZK. As noted in the previous section, we can restrict attention to deterministic verifiers V^* that get their bounded randomness as part of their bounded advice.

CLAIM 8. *Protocal* 9 *is ZK against any polynomial-time verifier* V^* *with advice of size at most* $b(n)$.

Proof sketch. We describe a universal (Remark 9) ZK simulator $\mathcal{S}$ and show its validity. Let $\varphi \in \mathcal{L}$, let V^* be the code of any malicious verifier, and let z' be any advice of length at most $b(n)$. $\mathcal{S}$ starts by running $V^*(\varphi; z')$, who returns $(y, \mathsf{c}, \mathsf{wi}_1)$, which are (allegedly) an image $f_n(x)$, an encryption of its preimage x, and the verifier message of the WI protocol.

The simulator $\mathcal{S}$ now constructs from the code of V^* a machine $\mathcal{M}_{V^*}$ that, given 1^n and $z = (z', \varphi)$ as input, outputs some y, and whose running time is linear in the running time t_{V^*} of V^*. In particular, $|z| \leq |z'| + |\varphi| \leq b(n) + n$. $\mathcal{S}$ then applies the extractor $\mathcal{E}$ on $\mathcal{M}_{V^*}$ and obtains a candidate witness $x' \in \{0,1\}^\ell$ in time $\mathrm{poly}(t^*_V)$. The simulator $\mathcal{S}$ now computes $\hat{\mathsf{c}} = \mathsf{Eval}(\mathcal{T}_{y,x'}, \mathsf{c})$, as well as a commitment C to x', and computes the second WI message wi_2 using the trapdoor x' as a witness for Ψ_2. It sends $(C, \hat{\mathsf{c}}, \mathsf{wi}_2)$ to complete the simulation.

The validity of the simulator now follows by witness-indistinguishability, as well as the circuit privacy guarantee. Specifically, we can first move to a hybrid experiment where the WI proof is given using the witness w. The view generated in this experiment is indistinguishable from the one generated by $\mathcal{S}$ due to the WI guarantee. Now, we claim that the view generated in this experiment is indistinguishable from that generated in a real interaction between the honest prover P and V^*. Indeed, there are two differences between the two. First, P commits to 0^ℓ instead of to x', and second, it sends $\hat{\mathsf{c}} \leftarrow \mathsf{Eval}(\mathbf{1}, \mathsf{c})$, instead of $\hat{\mathsf{c}} \leftarrow \mathsf{Eval}(\mathcal{T}_{y,x'}, \mathsf{c})$, for the extracted input x'. The two views are indistinguishable by the hiding of the commitment and by the function privacy guarantee of the 1-hop evaluation (this is argued exactly as in the proof of Claim 25).

This completes the proof sketch of Theorem 26. □

Appendix A. Black-box barriers. In our construction of (generalized) EOWFs against bounded-auxiliary-input adversaries, the extractor is non-black-box; namely, it makes explicit use of the adversary's code. In particular, the simulation of our 2-message and 3-message ZK protocols, which invokes this extractor, makes a non-black-box use of the adversarial verifier. In this section, we show that this is inherent by extending known results for adversaries with unbounded polynomial advice to the case of bounded-advice adversaries. We also observe that such black-box impossibilities do not hold for totally uniform adversaries (having no advice at all, on top of their constant size description).

EOWF with black-box extractors. We sketch why there do not exist EOWFs against b-bounded auxiliary-input adversaries where $b = n^{\Omega(1)}$, for security parameter n, and where the extractor only uses the adversary as a black box. Specifically, we show that given a function family $\mathcal{F}$ that satisfies one-wayness, there does not exist a PPT black box extractor $\mathcal{E}$ such that for any PPT adversary $\mathcal{M}$, any security parameter $n \in \mathbb{N}$, and any advice $z \in \{0,1\}^{b(n)}$,

$$\Pr_{e \leftarrow \mathcal{K}_{\mathcal{F}}(1^n)} \left[\begin{array}{c} y \leftarrow \mathcal{M}(e; z) \\ \exists x : f_e(x) = y \end{array} \wedge \begin{array}{c} x' \leftarrow \mathcal{E}^{\mathcal{M}(\cdot;z)}(e) \\ f_e(x') \neq y \end{array} \right] \leq \mathrm{negl}(n) \ .$$

This essentially follows the same idea behind the impossibility presented in section 4, only that now some of the computation done there by the obfuscated auxiliary input can be shifted from the auxiliary input to the adversary itself, as it is anyhow accessed as a black box. Concretely, consider the adversary $\mathcal{M}$ that interprets its auxiliary input z as a seed k of a PRF that maps the keys of $\mathcal{F}$ to inputs of $\mathcal{F}$. On input $(e; z)$, $\mathcal{M}$ computes an input $x = \mathsf{PRF}_z(e)$ and outputs $y = f_e(x)$. A similar argument to the one used in the proof of Theorem 9 in section 4 shows that any black-box extractor $\mathcal{E}$ can be used to break the one-wayness property of $\mathcal{F}$. (A similar barrier can be derived for the case of GEOWFs.)

Note that the above certainly does not hold when $b(n) = O(\log(n))$, since then the advice cannot contain a seed for a secure PRF. In fact, when $b(n) = O(\log(n))$, any family that is EOWF against b-bounded auxiliary-input adversaries also has a black-box extractor. The extractability property of the EOWF guarantees the existence of an extractor for every adversary $\mathcal{M}$ and advice z. Since there are only polynomially many different pairs $(\mathcal{M}, z)$, a black-box extractor can run the (possibly non-black-box) extractor for every such $(\mathcal{M}, z)$, and it is guaranteed that one of these executions outputs a valid preimage.

3-*message ZK with black-box simulation.* Goldreich and Krawczyk [GK96] show that a 3-message protocol for a language $\mathcal{L} \notin \mathrm{BPP}$ that is ZK against nonuniform verifiers cannot have a black-box simulator. That is, there is no simulator that only uses the verifier as a black box. To show this, they first construct a specific family $\mathcal{V}$ of nonuniform verifiers and then prove that any black-box simulator that can simulate verifiers in $\mathcal{V}$ can be used to decide $\mathcal{L}$ efficiently. This proof, however, does not directly rule out black-box simulation for bounded auxiliary-input verifiers. The reason is that, in the proof of [GK96], the advice given to verifiers in $\mathcal{V}$ encodes a key for a p-wise independent hash function where p bounds the running time of the simulator. Now, to rule out any polynomial-time simulator, we must require simulation for verifiers with advice of arbitrary polynomial length.

However, as observed in [BGGL01, PTW11], assuming one-way functions exist, one can replace the p-wise independent hash function in the construction of $\mathcal{V}$ by a PRF with seed length that is independent of p. Then, using the same argument as [GK96], one can show that black-box simulation is impossible even for b-bounded auxiliary-input verifiers where $b(n) = n^{\Omega(1)}$.

Similarly to the case of EOWF, there is no impossibility for 3-message ZK against verifiers with b-bounded auxiliary input where $b(n) = O(\log(n))$. In fact, as explained above, in this case, the non-black-box extractor of our GEOWF also implies a black-box extractor, which we can use to construct a black-box simulator in our 3-message ZK protocol.

2*-message ZK.* Goldreich and Oren [GO94] show that 2-message protocols for any language $\mathcal{L} \notin$ BPP that are ZK against nonuniform verifiers do not exist (even with non-black-box simulation). Their result crucially relies on the fact that the auxiliary input of the verifier can encode the first message of the protocol (and in particular extends to also rule out the case of bounded auxiliary-input verifiers, with advice longer that the first message). Our construction of 2-message ZK does not contradict the impossibility of [GO94] since it is only ZK against b-bounded auxiliary-input adversaries where b is smaller than the length of the first protocol message.

Acknowledgments. We thank Yael Kalai and Ron Rothblum for discussing the features of the [KRR14] delegation scheme. We also thank Ron for pointing out an inaccuracy, in an earlier version of this paper, regarding adaptive soundness via parallel repetition. We also thank the SICOMP referees for their valuable comments.

REFERENCES

[AIR01] W. Aiello, Y. Ishai, and O. Reingold, *Priced oblivious transfer: How to sell digital goods*, in Proceedings of EUROCRYPT, 2001, pp. 119–135.

[Bar01] B. Barak, *How to go beyond the black-box simulation barrier*, in Proceedings of FOCS, 2001, pp. 106–115.

[BC12] N. Bitansky and A. Chiesa, *Succinct arguments from multi-prover interactive proofs and their efficiency benefits*, in Proceedings of CRYPTO, 2012, pp. 255–272.

[BCC88] G. Brassard, D. Chaum, and C. Crépeau, *Minimum disclosure proofs of knowledge*, J. Comput. System Sci., 37 (1988), pp. 156–189.

[BCC+13] N. Bitansky, R. Canetti, A. Chiesa, S. Goldwasser, H. Lin, E. Tromer, and A. Rubinstein, *The Haunting of the Snark*, J. Cryptology (2016), pp. 1–78.

[BCC+14] N. Bitansky, R. Canetti, H. Cohn, S. Goldwasser, H. Lin, A. Rubinstein, and E. Tromer, *The Impossibility of Obfuscation with Auxiliary Input or a Universal Simulator*, CoRR abs/1401.0348, 2014.

[BCCT12] N. Bitansky, R. Canetti, A. Chiesa, and E. Tromer, *From extractable collision resistance to succinct non-interactive arguments of knowledge, and back again*, in Proceedings of the 3rd Innovations in Theoretical Computer Science Conference, ITCS '12, 2012, pp. 326–349.

[BCCT13] N. Bitansky, R. Canetti, A. Chiesa, and E. Tromer, *Recursive composition and bootstrapping for snarks and proof-carrying data*, in Proceedings of STOC, 2013, pp. 111–120.

[BCI+13] N. Bitansky, A. Chiesa, Y. Ishai, R. Ostrovsky, and O. Paneth, *Succinct non-interactive arguments via linear interactive proofs*, in Proceedings of TCC, 2013, pp. 315–333.

[BG08] B. Barak and O. Goldreich, *Universal arguments and their applications*, SIAM J. Comput., 38 (2008), pp. 1661–1694.

[BGGL01] B. Barak, O. Goldreich, S. Goldwasser, and Y. Lindell, *Resettably-sound zero-knowledge and its applications*, in Proceedings of FOCS, 2001, pp. 116–125.

[BGI+01] B. Barak, O. Goldreich, R. Impagliazzo, S. Rudich, A. Sahai, S. P. Vadhan, and K. Yang, *On the (im)possibility of obfuscating programs*, in Proceedings of CRYPTO, 2001, pp. 1–18.

[BGI14] E. Boyle, S. Goldwasser, and I. Ivan, *Functional signatures and pseudorandom functions*, in Proceedings of Public-Key Cryptography, PKC, 2014, 17th International Conference on Practice and Theory in Public-Key Cryptography, Buenos Aires, Argentina, 2014, pp. 501–519.

[BGL+15] N. Bitansky, S. Garg, H. Lin, R. Pass, and S. Telang, *Succinct randomized encodings and their applications*, in Proceedings of the 47th Annual ACM Conference on Symposium on Theory of Computing, STOC 2015, Portland, OR, 2015, pp. 439–448.

[BIN97] M. Bellare, R. Impagliazzo, and M. Naor, *Does parallel repetition lower the error in computationally sound protocols?*, in Proceedings of FOCS, 1997, pp. 374–383.

[Blu86] M. Blum, *How to prove a theorem so no one else can claim it*, in Proceedings of the International Congress of Mathematicians, 1986, pp. 1444–1451.

[BLV06] B. Barak, Y. Lindell, and S. P. Vadhan, *Lower bounds for non-black-box zero knowledge*, J. Comput. System Sci., 72 (2006), pp. 321–391.

[BP04a] M. Bellare and A. Palacio, *The knowledge-of-exponent assumptions and* 3*-round zero-knowledge protocols*, in Proceedings of the 24th Annual International Cryptology Conference, 2004, pp. 273–289.

[BP04b] M. Bellare and A. Palacio, *Towards plaintext-aware public-key encryption without random oracles*, in Proceedings of ASIACRYPT, 2004, pp. 48–62.

[BP13] N. Bitansky and O. Paneth, *On the impossibility of approximate obfuscation and applications to resettable cryptography*, in Proceedings of STOC, 2013, pp. 241–250.

[BP15] E. Boyle and R. Pass, *Limits of extractability assumptions with distributional auxiliary input*, in Proceedings of Advances in Cryptology, ASIACRYPT 2015, 21st International Conference on the Theory and Application of Cryptology and Information Security, Auckland, New Zealand, Part II, 2015, pp. 236–261,

[BSW12] D. Boneh, G. Segev, and B. Waters, *Targeted malleability: homomorphic encryption for restricted computations*, in Proceedings of ITCS, 2012, pp. 350–366.

[BV11] Z. Brakerski and V. Vaikuntanathan, *Efficient fully homomorphic encryption from (standard) LWE*, in Proceedings of FOCS, 2011, pp. 97–106.

[BW13] D. Boneh and B. Waters, *Constrained pseudorandom functions and their applications*, in Proceedings of Advances in Cryptology, ASIACRYPT 2013, 19th International Conference on the Theory and Application of Cryptology and Information Security, Bengaluru, India, Part II, 2013, pp. 280–300.

[CD08] R. Canetti and R. R. Dakdouk, *Extractable perfectly one-way functions*, in Proceedings of the 35th International Colloquium on Automata, Languages and Programming, 2008, pp. 449–460.

[CD09] R. Canetti and R. R. Dakdouk, *Towards a theory of extractable functions*, in Proceedings of TCC, 2009, pp. 595–613.

[CHJV15] R. Canetti, J. Holmgren, A. Jain, and V. Vaikuntanathan, *Succinct garbling and indistinguishability obfuscation for RAM programs*, in Proceedings of the 47th Annual ACM Conference on Symposium on Theory of Computing, STOC 2015, Portland, OR, 2015, pp. 429–437.

[CLP13] K.-M. Chung, H. Lin, and R. Pass, *Constant-round concurrent zero knowledge from p-certificates*, in Proceedings of FOCS, 2013.

[COSV12] C. Cho, R. Ostrovsky, A. Scafuro, and I. Visconti, *Simultaneously resettable arguments of knowledge*, in Proceedings of TCC, 2012, pp. 530–547.

[Dam92] I. Damgård, *Towards practical public key systems secure against chosen ciphertext attacks*, in Proceedings of CRYPTO91, 1992, pp. 445–456.

[DCL08] G. Di Crescenzo and H. Lipmaa, *Succinct NP proofs from an extractability assumption*, in Proceedings of the 4th Conference on Computability in Europe, 2008, pp. 175–185.

[DFH12] I. Damgård, S. Faust, and C. Hazay, *Secure two-party computation with low communication*, in Proceedings of TCC, 2012, pp. 54–74.

[DN07] C. Dwork and M. Naor, *Zaps and their applications*, SIAM J. Comput., 36 (2007), pp. 1513–1543.

[FLS99] U. Feige, D. Lapidot, and A. Shamir, *Multiple noninteractive zero knowledge proofs under general assumptions*, SIAM J. Comput., 29 (1999), pp. 1–28.

[FS90] U. Feige and A. Shamir, *Witness indistinguishable and witness hiding protocols*, in Proceedings of STOC, 1990, pp. 416–426.

[GGH13a] S. Garg, C. Gentry, and S. Halevi, *Candidate multilinear maps from ideal lattices*, in Proceedings of EUROCRYPT, 2013, pp. 1–17.

[GGH$^+$13b] S. Garg, C. Gentry, S. Halevi, M. Raykova, A. Sahai, and B. Waters, *Candidate indistinguishability obfuscation and functional encryption for all circuits*, in Proceedings of FOCS, 2013.

[GGM86] O. Goldreich, S. Goldwasser, and S. Micali, *How to construct random functions*, J. ACM, 33 (1986), pp. 792–807.

[GGPR13] R. Gennaro, C. Gentry, B. Parno, and M. Raykova, *Quadratic span programs and succinct nizks without pcps*, in Proceedings of EUROCRYPT, 2013, pp. 626–645.

[GHV10] C. Gentry, S. Halevi, and V. Vaikuntanathan, *i-hop homomorphic encryption and rerandomizable yao circuits*, in Proceedings of CRYPTO, 2010, pp. 155–172.

[GK96] O. Goldreich and H. Krawczyk, *On the composition of zero-knowledge proof systems*, SIAM J. Comput., 25 (1996), pp. 169–192.

[GK05] S. Goldwasser and Y. T. Kalai, *On the impossibility of obfuscation with auxiliary input*, in Proceedings of FOCS, 2005, pp. 553–562.

[GLR11] S. Goldwasser, H. Lin, and A. Rubinstein, *Delegation of Computation Without Rejection Problem from Designated Verifier CS-Proofs*, Cryptology ePrint Archive, Report 2011/456, 2011.

[GMR89] S. Goldwasser, S. Micali, and C. Rackoff, *The knowledge complexity of interactive proof systems*, SIAM J. Comput., 18 (1989), pp. 186–208.

[GMW87] O. Goldreich, S. Micali, and A. Wigderson, *How to play any mental game*, in STOC '87: Proceedings of the 19th Annual ACM Symposium on Theory of Computing, 1987, pp. 218–229.

[GO94] O. Goldreich and Y. Oren, *Definitions and properties of zero-knowledge proof systems*, J. Cryptology, 7 (1994), pp. 1–32.

[Gol93] O. Goldreich, *A uniform-complexity treatment of encryption and zero-knowledge*, J. Cryptology, 6 (1993), pp. 21–53.

[Gol04] O. Goldreich, *Foundations of Cryptography: Volume* 2*, Basic Applications*, Cambridge University Press, New York, 2004.

[Gro10] J. Groth, *Short pairing-based non-interactive zero-knowledge arguments*, in Proceedings of the 16th International Conference on the Theory and Application of Cryptology and Information Security, 2010, pp. 321–340.

[GW11] C. Gentry and D. Wichs, *Separating succinct non-interactive arguments from all falsifiable assumptions*, in Proceedings of the 43rd Annual ACM Symposium on Theory of Computing, 2011, pp. 99–108.

[HK12] S. Halevi and Y. T. Kalai, *Smooth projective hashing and two-message oblivious transfer*, J. Cryptology, 25 (2012), pp. 158–193.

[HT98] S. Hada and T. Tanaka, *On the existence of* 3*-round zero-knowledge protocols*, in Proceedings of the 18th Annual International Cryptology Conference, 1998, pp. 408–423.

[KLW15] V. Koppula, A. B. Lewko, and B. Waters, *Indistinguishability obfuscation for turing machines with unbounded memory*, in Proceedings of the 47th Annual ACM Conference on Symposium on Theory of Computing, STOC 2015, Portland, OR, 2015, pp. 419–428.

[KMN$^+$14] I. Komargodski, T. Moran, M. Naor, R. Pass, A. Rosen, and E. Yogev, *One-way functions and (im)perfect obfuscation*, in Proceedings of the 55th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2014, Philadelphia, 2014, pp. 374–383.

[KPTZ13] A. Kiayias, S. Papadopoulos, N. Triandopoulos, and T. Zacharias, *Delegatable pseudorandom functions and applications*, in Proceedings of the ACM SIGSAC Conference on Computer and Communications Security, CCS'13, Berlin, 2013, pp. 669–684, http://doi.acm.org/10.1145/2508859.2516668.

[KRR14] Y. T. Kalai, R. Raz, and R. D. Rothblum, *How to delegate computations: The power of no-signaling proofs*, in Proceedings of STOC, 2014.

[Lip12] H. Lipmaa, *Progression-free sets and sublinear pairing-based non-interactive zero-knowledge arguments*, in Proceedings of TCC, 2012, pp. 169–189.

[Lip13] H. Lipmaa, *Succinct non-interactive zero knowledge arguments from span programs and linear error-correcting codes*, in Proceedings of Advances in Cryptology, ASIACRYPT 2013, 19th International Conference on the Theory and Application of Cryptology and Information Security, Bengaluru, India, Part I, 2013, pp. 41–60.

[LS90] D. Lapidot and A. Shamir, *Publicly verifiable non-interactive zero-knowledge proofs*, in Proceedings of CRYPTO, 1990, pp. 353–365.

[Mic00] S. Micali, *Computationally sound proofs*, SIAM J. Comput., 30 (2000), pp. 1253–1298.

[Mie08] T. Mie, *Polylogarithmic two-round argument systems*, J. Math. Cryptology, 2 (2008), pp. 343–363.

[Nao03] M. Naor, *On cryptographic assumptions and challenges*, in Proceedings of the 23rd Annual International Cryptology Conference, 2003, pp. 96–109.

[NP01] M. Naor and B. Pinkas, *Efficient oblivious transfer protocols*, in Proceedings of SODA, 2001, pp. 448–457.

[OV12] R. Ostrovsky and I. Visconti, *Simultaneous resettability from collision resistance*, Electronic Colloquium on Computational Complexity (ECCC), 2012.

[PTW11] R. Pass, W. D. Tseng, and D. Wikström, *On the composition of public-coin zero-knowledge protocols*, SIAM J. Comput., 40 (2011), pp. 1529–1553.

[PW09] R. Pass and H. Wee, *Black-box constructions of two-party protocols from one-way functions*, in Proceedings of TCC, 2009, pp. 403–418.

[SW14] A. Sahai and B. Waters, *How to use indistinguishability obfuscation: Deniable encryption, and more*, in Proceedings of STOC, 2014.

[Yao86] A. C.-C. Yao, *How to generate and exchange secrets (extended abstract)*, in Proceedings of FOCS, 1986, pp. 162–167.