On the Existence of Extractable One-Way Functions[*]

Nir Bitansky [†]
Tel Aviv University

Ran Canetti [‡]
Boston University and Tel Aviv University

Omer Paneth[§]
Boston University

Alon Rosen[¶]
Efi Arazi School of Computer Science,
IDC Herzliya, Israel

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 non-standard *knowledge assumption* on certain functions.

We make two headways in the study of the existence of extractable one-way functions (EOWFs). On the negative side, we show that if there exist indistinguishability obfuscators for a certain class of 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 (e.g., sub-exponential 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 the same class of adversarial verifiers, from essentially the same assumptions.

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 are an obvious example for the use of knowledge extraction. In fact, here 'knowledge' is *defined* by way of existence of an efficient extraction procedure. 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 or multi-party computation [GMR89, GMW87]. A quintessential example here is the Feige-Lapidot-Shamir paradigm [FLS99]. Other contexts are mentioned within.

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 and does not need to know anything regarding its "internals". However, as a number of impossibility results for black-box reductions and simulation show, this technique is also quite limited. One main limitation of rewinding-based extraction is that it requires multiple rounds of interaction with the adversary. Indeed, proving security of candidate 3-message zero-knowledge protocols, succinct non-interactive arguments (SNARGs), and other tasks are out of the technique's reach [GK96, GW11].

Starting with the work of Barak et al. [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, thus 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 out-

[*] 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).

[†] Supported by an IBM Ph.D. Fellowship and the Check Point Institute for Information Security.

[‡] Supported by the Check Point Institute for Information Security, an NSF EAGER grant, and NSF Algorithmic foundations grant 1218461.

[§] Supported by the Simons award for graduate students in theoretical computer science and an NSF Algorithmic foundations grant 1218461.

[¶] Supported by ISF grant no. 1255/12 and by the ERC under the EU's Seventh Framework Programme (FP/2007-2013) ERC Grant Agreement n. 307952.

puts y in the image of f_e has an "extractor" $\mathcal{E}$ that given e and the code of $\mathcal{A}$, outputs a preimage of y.

Extractable functions provide an alternative (albeit non-explicit) "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 [CD09, BCC$^+$13], are known to suffice for constructing 3-message zero-knowledge protocols, and extractable collision-resistant hash functions are known to suffice for constructing succinct non-interactive arguments (SNARGs) [BCCT12]. KEA had also led to relatively efficient CCA constructions [Dam92, BP04a].

The black-box impossibility of some of the above applications imply 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 the suitable non-black-box techniques has been the main obstacle in constructing extractable functions, and to date, no construction with an explicit extraction procedure is known. Instead, for all the 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 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 partial.

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 algorithm) there exists an extractor (again, modeled as a uniform 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 non-uniform families of deterministic polysize 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's 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 polysize 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. In the above, we can either model both the adversary $\mathcal{A}$ and the extractor $\mathcal{E}$ as uniform polytime machines, or as non-uniform machines with polynomial size advice. We call z the *common auxiliary input*, and if $\mathcal{A}$ and $\mathcal{E}$ are non-uniform 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 zero-knowledge protocols remain zero-knowledge under sequential composition, the verifier and simulator get common auxiliary input. Indeed, to obtain this standard formulation of zero-knowledge 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, see e.g. [BCCT12].

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 extractable one-way functions with common auxiliary input of *unbounded* length may not exist:

THEOREM 1 (INFORMAL). *If there exist indistinguishability obfuscators for a certain class of circuits, then there do not exist extractable one-way functions with respect to common auxiliary-input of unbounded polynomial length.*

This seems to suggest that the concept of extractable one-way functions (and other concepts that imply it, such as extractable collision-resistant hashing or SNARKs) may be shaky overall, especially in light of the recent candidate indistinguishability obfuscator for all circuits [GGH$^+$13b].

Still, 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 publicly-verifiable P-delegation schemes:

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

While the existence of publicly-verifiable P-delegation schemes is perhaps not considered as a standard assumption, it is a *falsifiable* assumption [Nao03],[1] with candidates such as CS proofs [Mic00] or SNARGs [BCCT13] (when restricted to P). We view this construction mainly as a proof of concept, showing that ruling out such extractable functions may be a hard task.

Trying to head towards a construction from standard assumptions, we formulate a generalized variant of extractable one-way functions (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 P-delegation, which was recently established by [**?**], 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 zero-knowledge arguments and 3-message zero-knowledge arguments of knowledge, against verifiers with bounded-auxiliary-input.

THEOREM 3 (INFORMAL).

1. *Assuming (even privately-verifiable) P-delegation, there exist GEOWFs with respect to (common or individual) auxiliary-input of bounded polynomial length.*

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

2. *Assuming GEOWFs, ZAPs [DN07], and (even 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 adversaries, there exists a 2-message ZK argument against bounded-auxiliary-input verifiers.*

We now elaborate on each of the results.

1.2 Impossibility with respect to Unbounded Auxiliary-Input

To introduce the negative result regarding EOWFs with unbounded (common) auxiliary-input, we first recall the notion of obfuscation, and explain their contrast with auxiliary-input extractability.

Obfuscation. Program obfuscation is aimed at making code unintelligible while preserving its functionality, and has been long considered to be a holy grail of cryptography, with diverse and far reaching applications. Barak et al. [BGI$^+$01] initiated the rigourous 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$^+$13b] who proposed a candidate construction of general-purpose obfuscators. They show that, under algebraic assumptions closely related to multilinear maps [GGH13a, CLT13], their construction satisfies the relaxed notion of *indistinguishability obfuscation* (IO) [BGI$^+$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.

Since the emergence of the Garg et al. candidate, IO has been shown to have variety of powerful positive applications, such as functional encryption, public-key encryption from one way functions, deniable encryption, 2-message multi-party computation, and more [GGH$^+$13b, SW13, HSW13, GGHR13, BZ13, KRW13].

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, **?**]. We show that general 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 for example 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 a pseudo-random function PRF_k 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 pseudo-randomness 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.[2] Nevertheless, we show that the above intuition can still be fulfilled. The idea is to consider an alternative to the 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 [SW13].

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 pseudo-random, even given (x^*, k_{x^*}). As shown in several recent works [BW13, BGI13, 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 the 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 indistinguishability obfuscation. However, now by the pseudo-randomness guarantee at the punctured point e, we know that $\mathsf{PRF}_k(e)$ is pseudo random, 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 OWF, it follows that the impossibility of EOWFs is implied by indistinguishability obfuscation 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 knowledge of exponent assumption wrong? In its original formulation [Dam92] and in subsequent formulations [HT98, BP04a, BP04b], the knowledge of exponent assumption (KEA) was not stated with respect to common auxiliary-input, but rather only for individual auxiliary-input (or completely uniform machines), 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 non-uniform 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

[2] In fact, formalizing the above intuition is tricky even with VBB, because one has to reduce extraction of one of perhaps many arbitrary pre-images to the task of predicting some deterministic predicate of the PRF key k.

extraction (even assuming IO for all circuits).

Our result does not disvalidate 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", e.g. the uniform distribution, required in [BCCT12].

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 [BP13b], 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* (a.k.a. *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 ours and Boyle and Pass' 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 a generalized version of EOWFs (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 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 key-less, it is simply a pseudorandom generator 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 pseudo-randomness 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 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 for constructing 2-message and 3-message ZK protocols.

A similar problem is encountered in Barak's zero-knowledge protocol [Bar01], where the entire computation of a malicious verifier is used as the simulation trapdoor. As in the 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 simply computing a proof for its computation.

P-delegation. The proof system required in our constructions is a non-interactive computationally sound proof for deterministic polytime statements, from hereon referred to as a P-delegation scheme. More precisely, in a P-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 non-interactive 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 P-delegation, their security is not based on standard assumptions. In a recent breakthrough result, Kalai, Raz and Rothblum [**?**] construct a privately verifiable P-delegation scheme based on any private information retrieval scheme with sub-exponential security. While the scheme of [**?**] only has non-adaptive soundness, we use standard techniques to get soundness for a statement that is adaptively chosen from a relatively small set of possible statements. This is indeed what is required for our construction (see the body for more details).

GEOWF from P-delegation. We now sketch how P-delegation is used in our constructions. We obtain publicly-verifiable (respectively, privately-verifiable) GEOWFs based on publicly-verifiable (respectively, privately-verifiable) delegation. In both cases, the GEOWF f is key-less, it is given as input a seed s and a random string r. f applies a pseudo-random generator 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 P-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 inefficiently computed from σ. (Both assumption are WLOG.)

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 then 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 $i \neq 0^n$, the output is computed in the "trapdoor branch", which is 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 so 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 body, we show how to avoid this problem by relying on the specific construction of publicly-verifiable GEOWFs from publicly-verifiable P-delegation with 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 zero-knowledge protocols from standard (non-generalized) EOWFs, and then explain how to replace the EOWFs with GEOWFs.

From EOWF to 3-message zero knowledge. The protocol follows the Feige-Lapidot-Shamir *trapdoor paradigm* [FLS99]. Given, say a key-less, 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 proof-of-knowledge attesting that it either knows a witness w for the proven statement, or it 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 WI proof of knowledge 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 only determined when the verifier sends y in the second protocol message. Therefore, we must make sure to use a WI proof of knowledge 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 an advice, we must therefore 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 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 zero knowledge. In the 2-message protocol, we replace the 3-message WI proof of knowledge with a 2-message WI proof (e.g. a ZAP). However, in the above 3-message protocol, soundness is established by using the proof-of-knowledge property of the WI, whereas 2-message WI proofs of knowledge are not known. Instead, we prove soundness using complexity leveraging. The prover adds to its message a statistically-binding commitment to junk, and proves that either "$x \in \mathcal{L}$", or "$f(x) = y$ *and the commitment is 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 $\mathrm{poly}(T)$. 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 zero-knowledge protocols on privately-verifiable GEOWFs (that 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 WI proof of knowledge attesting that it either 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 two-message secure function evaluation protocol, based for example on a 1-hop homomorphic encryption [GHV10]. More concretely, the verifier, in addition to the the function 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 WI proof of knowledge attesting that it either 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 two and three message ZK and related work. three-message zero-knowledge 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 (for more details, see the full version of this paper). Previous three-message zero-knowledge protocols were based either on the knowledge of exponent assumption [HT98, BP04a], on extractable one-way functions[BCC$^+$13], or on other extractability assumptions [CD08]. In all, the simulator uses a non-black extractor that is only assumed to exist, but not explicitly constructed.

Two-message zero-knowledge arguments against adversaries with unbounded polynomial advice exist only for trivial languages (regardless of how simulation is done) [GO94]. In fact, 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 two-message argument that is zero-knowledge as long as the verifier's advice is shorter than the verifier message by super-logarithmic additive factor. Indeed, our two-message protocol has the same skeleton. However, security of the Barak et al. protocol is only shown assuming existence of P-delegation schemes (or universal arguments for non-deterministic 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 function (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, extractable collision-resistant hash-functions and extractable injective one-way functions.

3. Finally, we ask whether there exist EOWF's with respect to common auxiliary input that is taken from specific "benign" distribution, such as the uniform distribution.

Organization

In Section 2, we give the relevant definitions for EOWF and GEOWF. In Section 3, we present the limitation on unbounded auxiliary-input EOWFs based on indistinguishability obfuscation. In Section 4, we present the constructions of bounded-auxiliary-input EOWFs and GEOWFs. The constructions of zero-knowledge protocols from GEOWFs and a discussion of relevant black-box lower for EOWFs and ZK are given in the full version of this paper.

2. EXTRACTABLE ONE-WAY FUNCTIONS

In this section, we define auxiliary-input extractable one-way functions (EOWFs), bounded-auxiliary-input EOWFs, and generalized extractable one-way functions (GEOWFs).

DEFINITION 1 (AUXILIARY-INPUT EOWFS [CD08]). *Let ℓ, ℓ', m be polynomially bounded length functions. An efficiently computable family of functions*

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

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

1. **One-way:** *For any PPT $\mathcal{A}$, polynomial b, large enough security parameter $n \in \mathbb{N}$, and $z \in \{0,1\}^{b(n)}$:*

$$\Pr_{\substack{e \leftarrow \mathcal{K}_{\mathcal{F}}(1^n) \\ x \leftarrow \{0,1\}^{\ell(n)}}} \left[\begin{array}{c} x' \leftarrow \mathcal{A}(e, f_e(x); z) \\ f_e(x') = f_e(x) \end{array} \right] \leq \operatorname{negl}(n) .$$

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

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

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 only guaranteed 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 polytime adversaries*.

For b-bounded auxiliary input, we also define key-less families. While for unbounded auxiliary input, extraction is impossible for key-less 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 key-less functions.

DEFINITION 2 (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*

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

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

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

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

We say that the function is **key-less** *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 against bounded auxiliary-input verifiers, 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 vs. 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 zero-knowledge or obfuscation and, as explained in the intro, is instrumental when arguing about extractable functions in the context of a larger system. As also mentioned in the intro, 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 CCA and plaintext-aware encryption with non-uniform 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 includes essentially all the works on succinct non-interactive arguments (SNARGs), succinct NIZKs, and targeted malleability, that rely on extractable primitives [DCL08, Mie08, Gro10, GLR11, BSW12, BCCT12, BC12, DFH12, Lip12, BCCT13, BCI$^+$13, GGPR13, Lip13].*

2.1 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 only gets 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, can find a preimage x'. *Generalized EOWFs* (GEOWFs) allows to express this asymmetry in a more flexible way. Concretely, a function family $\mathcal{F}$ is now associated with a "hard" relation $\mathcal{R}^{\mathcal{F}}_e(f_e(x), x')$ 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 $\mathcal{R}^{\mathcal{F}}_e(y, x') = 1$. 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 $\mathcal{R}^{\mathcal{F}}_e(y, x)$, can be efficiently tested given y and x only (and the key e if the function is keyed). The second is *privately-verifiable GEOWFs*, where the relation $\mathcal{R}^{\mathcal{F}}_e(y, x)$, might not be efficiently testable given only $(y = f_e(x'), x)$, but is possible to efficiently test the relation given x' in addition.

We note that standard EOWFs, as given in Definition 1, fall under the category of publicly-verifiable GEOWFs, where the relation $\mathcal{R}^{\mathcal{F}}_e(y, x)$ simply tests whether $y = f_e(x)$.

DEFINITION 3 (GEOWFS). *An efficiently computable family of functions*

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

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

1. **$\mathcal{R}^{\mathcal{F}}$-Hard:** *For any PPT $\mathcal{A}$, polynomial b, large enough security parameter $n \in \mathbb{N}$, and $z \in \{0,1\}^{b(n)}$:*

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

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, large enough security parameter $n \in \mathbb{N}$, and $z \in \{0,1\}^{b(n)}$:*

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

We further say that the function is

- **Publicly-verifiable** *if $\mathcal{R}^{\mathcal{F}}_e(f_e(x), x')$ can always be efficiently computed by a tester $\mathcal{T}(e, f_e(x), x')$.*
- **Privately-verifiable** *if $\mathcal{R}^{\mathcal{F}}_e(f_e(x), x')$ can be efficiently computed by a tester a tester $\mathcal{T}(e, x, x')$.*

Bounded auxiliary input GEOWFs (b-bounded auxiliary-input GEOWFs) are defined analogously to b-bounded auxiliary-input-EOWFs. That is, $\mathcal{R}^{\mathcal{F}}$-hardness is defined exactly as in Definition 3, whereas $\mathcal{R}^{\mathcal{F}}$-hardness is only against adversaries with auxiliary input of an apriori fixed polynomial size $b(n)$.

3. FROM IO TO IMPOSSIBILITY OF UNBOUNDED-AUXILIARY-INPUT EOWFS

We show that if there exists indistinguishability obfuscation (IO), there do not exist (generalized) auxiliary-input extractable one-way functions.

THEOREM 4. *Assuming indistinguishability obfuscation for all circuits, neither EOWFs nor GEOWFs exist, with respect to common auxiliary-input of unbounded polynomial length.*

Due to lack of space, the relevant definition and the proof of Theorem 4 are given in the full version of this paper.

REMARK 3 (OTHER EXTRACTABLE PRIMITIVES). *GEOWFs are a minimal extractable cryptographic primitive, in the sense that other extractable primitives such as extractable collision-resistance hash functions (ECRHs), or succinct non-interactive arguments of knowledge (SNARKs) imply them. (For example, in [BCCT12], it is shown that SNARKs imply proximity ECRHs, which in turn 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.*

4. BOUNDED-AUXILIARY-INPUT EXTRACTABLE ONE-WAY FUNCTIONS

In this section, we construct bounded-auxiliary-input extractable one-way functions (EOWFs) and bounded auxiliary-input-generalized EOWFs (GEOWFs). Before presenting the construction, we define *non-interactive universal arguments for deterministic computations*, which is the main tool we rely on, and discuss an instantiation based on the delegation scheme of Kalai, Raz, and Rothblum [**?**].

4.1 Non-Interactive Universal Arguments for Deterministic Computations & Delegation

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 $T(n) \in (2^{\omega(\log n)}, 2^{\text{poly}(n)})$ be a computable superpolynomial function. An NIUA 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, an NIUA system is defined as follows.

DEFINITION 4 (NIUA). *A triple $(\mathcal{G}, \mathcal{P}, \mathcal{V})$ is a non-interactive universal argument system for for Dtime(T) if it satisfies:*

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 polysize prover $\mathcal{P}^*$, large enough $n \in \mathbb{N}$, and set of at most $2^{b(n)}$ false statements $S \subseteq \{0,1\}^{\text{poly}(n)} \setminus \mathcal{L}_{\mathcal{U}}(T(n))$:*

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

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 WLOG that the verification state τ appears in the clear in the reference string σ.*

THEOREM 5 (FOLLOWING FROM [**?**]). *Assuming the Learning with Errors Problem is sub-exponentially 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) NIUA with adaptive soundness for $2^{b(n)}$ statements.*

4.2 Constructions

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

4.2.1 The generalized extractable one-way function

Let $b(n)$ be a polynomial. Let $(\mathcal{G}, \mathcal{P}, \mathcal{V})$ be an NIUA system 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 $p(n)$. Let PRG be a pseudo random generator stretching n bits to $b(n) + n$ bits. We construct a key-less family of functions $\mathcal{F} = \{f_n\}_{n \in \mathbb{N}}$, consisting of one function $f_n : \{0,1\}^{\ell(n)} \rightarrow \{0,1\}^{\ell'(n)}$, 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 1, and is followed by the corresponding relation $\mathcal{R}^{\mathcal{F}}$.

Inputs: (s, r, pad) of respective lengths $(n, n, \ell(n) - 2n)$.

1. Compute $v = \mathsf{PRG}(s)$.
2. Sample NIUA reference string and verification state $(\sigma, \tau) \leftarrow \mathcal{G}(1^n; r)$.
3. Output (σ, v).

Figure 1: The function f_n.

We now define the corresponding relation $\mathcal{R}^{\mathcal{F}} = \left\{\mathcal{R}_n^{\mathcal{F}}\right\}_{n \in \mathbb{N}}$ in Figure 2, which will be publicly-verifiable (respectively, privately-verifiable) if the NIUA is publicly (respectively, privately verifiable). For simplicity, we assume that the NIUA 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, NIUA verification can be done efficiently, and thus the relation $\mathcal{R}_n^{\mathcal{F}}$ can be efficiently tested.

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

REMARK 4 (ONE-WAYNESS VS. $\mathcal{R}^{\mathcal{F}}$-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*

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. Compute the (unique) verification state τ corresponding to the reference string σ:
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.

Figure 2: The relation $\mathcal{R}_n^{\mathcal{F}}(f_n(x), x')$.

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 6. *The function family $\mathcal{F} = \{f_n\}_{n\in\mathbb{N}}$, given in Figure 1 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 NIUA.

As for extraction, given a poly-time machine $\mathcal{M}_z$ with short advice z that outputs (σ, v), where σ is a valid reference string for the NIUA system, 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 NIUA system, the proof π is indeed accepting, and the witness satisfies $\mathcal{R}^{\mathcal{F}}$. Furthermore, by the relative prover efficiency of the NIUA, the extractor runs in time that is polynomial in the running time of the adversary $\mathcal{M}_z$.

The full proof of Theorem 6 is given in the full version of this paper.

4.2.2 The standard extractable one-way function

We construct a standard extractable one-way function based on publicly-verifiable NIUAs 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 exiting candidates for publicly-verifiable NIUAs indeed have this property.[3]

DEFINITION 5 (NIUA WITH KEY VALIDATION). *A publicly-verifiable NIUA system is said to have key validation if there exists an efficient 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, which at a very high-level attempts to embed the structure of the previous GEOWF function and relation into a standard EOWF.

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

We construct a key-less 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 3.

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 (v^*, σ^*).
- If $i = 0^n$:
  1. Perform the following tests:
     - Run $\mathsf{Valid}(\sigma)$ to check the validity of σ,
     - 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 (v, σ).
  2. Otherwise, output $\perp$.
- If $i = 1^n$, output $\perp$.

Figure 3: The function f_n.

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

THEOREM 7. *The function family $\mathcal{F} = \{f_n\}_{n\in\mathbb{N}}$, given in Figure 3 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 NIUA as in the proof of Theorem 6 (leveraging the fact that a truly random $\tilde{v}$ almost surely cannot be computed by a short machine).

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

As for extraction, given a poly-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 NIUA system (indeed, $\perp$ is an image that indicates an improper reference string σ, or a non-accepting 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 NIUA system, for a valid σ, the proof π is indeed accepting. By the relative prover efficiency of the NIUA, 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 $i = 1^n$.

The full proof of Theorem 7 is given in the full version of this paper.

Acknowledgements

We thank Yael Kalai and Ron Rothblum for discussing the features of the **[?]** delegation scheme. We also thank Ron for pointing out an inaccuracy, in an earlier version of this paper, regarding adaptive soundness via parallel repetition.

5. REFERENCES

[1] Barak, B. How to go beyond the black-box simulation barrier. In *FOCS* (2001), pp. 106–115.

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

[3] Barak, B., Lindell, Y., and Vadhan, S. P. Lower bounds for non-black-box zero knowledge. *J. Comput. Syst. Sci. 72*, 2 (2006), 321–391.

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

[5] Bellare, M., and Palacio, A. Towards plaintext-aware public-key encryption without random oracles. In *ASIACRYPT* (2004), pp. 48–62.

[6] Bitansky, N., Canetti, R., Chiesa, A., Goldwasser, S., Lin, H., Tromer, E., and Rubinstein, A. The haunting of the snark. *Manuscript* (2013).

[7] Bitansky, N., Canetti, R., Chiesa, A., and Tromer, E. 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* (2012), ITCS '12, pp. 326–349.

[8] Bitansky, N., Canetti, R., Chiesa, A., and Tromer, E. Recursive composition and bootstrapping for snarks and proof-carrying data. In *STOC* (2013), pp. 111–120.

[9] Bitansky, N., Canetti, R., Cohn, H., Goldwasser, S., Kalai, Y. T., Paneth, O., and Rosen, A. The impossibility of obfuscation with auxiliary input or a universal simulator. *CoRR abs/1401.0348* (2014).

[10] Bitansky, N., and Chiesa, A. Succinct arguments from multi-prover interactive proofs and their efficiency benefits. In *CRYPTO* (2012), pp. 255–272.

[11] Bitansky, N., Chiesa, A., Ishai, Y., Ostrovsky, R., and Paneth, O. Succinct non-interactive arguments via linear interactive proofs. In *TCC* (2013), pp. 315–333.

[12] Boneh, D., Segev, G., and Waters, B. Targeted malleability: homomorphic encryption for restricted computations. In *ITCS* (2012), pp. 350–366.

[13] Boneh, D., and Waters, B. Constrained pseudorandom functions and their applications. *IACR Cryptology ePrint Archive 2013* (2013), 352.

[14] Boneh, D., and Zhandry, M. Multiparty key exchange, efficient traitor tracing, and more from indistinguishability obfuscation. *IACR Cryptology ePrint Archive 2013* (2013), 642.

[15] Boyle, E., Goldwasser, S., and Ivan, I. Functional signatures and pseudorandom functions. *IACR Cryptology ePrint Archive 2013* (2013), 401.

[16] Boyle, E., and Pass, R. Limits of extractability assumptions with distributional auxiliary input. *IACR Cryptology ePrint Archive 2013* (2013), 703.

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

[18] Canetti, R., and Dakdouk, R. R. Towards a theory of extractable functions. In *TCC* (2009), pp. 595–613.

[19] Chung, K.-M., Lin, H., and Pass, R. Constant-round concurrent zero knowledge from p-certificates. In *FOCS* (2013).

[20] Coron, J.-S., Lepoint, T., and Tibouchi, M. Practical multilinear maps over the integers. In *CRYPTO (1)* (2013), pp. 476–493.

[21] Damgård, I. Towards practical public key systems secure against chosen ciphertext attacks. In *Proceedings of CRYPTOâĂŹ91* (1992), pp. 445–456.

[22] Damgård, I., Faust, S., and Hazay, C. Secure two-party computation with low communication. In *TCC* (2012), pp. 54–74.

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

[24] Dwork, C., and Naor, M. Zaps and their applications. *SIAM J. Comput. 36*, 6 (2007), 1513–1543.

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

[26] Garg, S., Gentry, C., and Halevi, S. Candidate multilinear maps from ideal lattices. In *EUROCRYPT* (2013), pp. 1–17.

[27] Garg, S., Gentry, C., Halevi, S., and Raykova, M. Two-round secure mpc from indistinguishability obfuscation. *IACR Cryptology ePrint Archive 2013* (2013), 601.

[28] Garg, S., Gentry, C., Halevi, S., Raykova, M., Sahai, A., and Waters, B. Candidate indistinguishability obfuscation and functional encryption for all circuits. In *FOCS* (2013).

[29] Gennaro, R., Gentry, C., Parno, B., and Raykova, M. Quadratic span programs and succinct nizks without pcps. In *EUROCRYPT* (2013), pp. 626–645.

[30] Gentry, C., Halevi, S., and Vaikuntanathan, V. *i*-hop homomorphic encryption and rerandomizable yao circuits. In *CRYPTO* (2010), pp. 155–172.

[31] Gentry, C., and Wichs, D. 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.

[32] Goldreich, O., Goldwasser, S., and Micali, S. How to construct random functions. *J. ACM 33*, 4 (1986), 792–807.

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

[34] Goldreich, O., Micali, S., and Wigderson, A. How to play any mental game. In *STOC '87: Proceedings of the nineteenth annual ACM symposium on Theory of computing* (1987), pp. 218–229.

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

[36] Goldwasser, S., and Kalai, Y. T. On the impossibility of obfuscation with auxiliary input. In *FOCS* (2005), pp. 553–562.

[37] Goldwasser, S., Lin, H., and Rubinstein, A. Delegation of computation without rejection problem from designated verifier CS-proofs. Cryptology ePrint Archive, Report 2011/456, 2011.

[38] Goldwasser, S., Micali, S., and Rackoff, C. The knowledge complexity of interactive proof systems. *SIAM J. Comput. 18*, 1 (1989), 186–208.

[39] Groth, J. Short pairing-based non-interactive zero-knowledge arguments. In *ASIACRYPT* (2010), pp. 321–340.

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

[41] Hohenberger, S., Sahai, A., and Waters, B. Replacing a random oracle: Full domain hash from indistinguishability obfuscation. *IACR Cryptology ePrint Archive 2013* (2013), 509.

[42] Kalai, Y. T., Raz, R., and Rothblum, R. D. How to delegate computations: The power of no-signaling proofs. In *STOC* (2014).

[43] Kiayias, A., Papadopoulos, S., Triandopoulos, N., and Zacharias, T. Delegatable pseudorandom functions and applications. *IACR Cryptology ePrint Archive 2013* (2013), 379.

[44] Koppula, V., Ramchen, K., and Waters, B. Separations in circular security for arbitrary length key cycles. *IACR Cryptology ePrint Archive 2013* (2013), 683.

[45] Lipmaa, H. Progression-free sets and sublinear pairing-based non-interactive zero-knowledge arguments. In *TCC* (2012), pp. 169–189.

[46] Lipmaa, H. Succinct non-interactive zero knowledge arguments from span programs and linear error-correcting codes. *IACR Cryptology ePrint Archive 2013* (2013), 121.

[47] Micali, S. Computationally sound proofs. *SIAM Journal on Computing 30*, 4 (2000), 1253–1298. Preliminary version appeared in FOCS '94.

[48] Mie, T. Polylogarithmic two-round argument systems. *Journal of Mathematical Cryptology 2*, 4 (2008), 343–363.

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

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

[51] Sahai, A., and Waters, B. How to use indistinguishability obfuscation: Deniable encryption, and more. *IACR Cryptology ePrint Archive 2013* (2013), 454.