On the Existence of Extractable One-Way Functions*

Nir Bitansky† Ran Canetti‡ Omer Paneth§ Alon Rosen¶

May 31, 2014

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.

*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 STOC 2014.

†Tel Aviv University. Email: nirbitan@tau.ac.il. Supported by an IBM Ph.D. Fellowship and the Check Point Institute for Information Security.

‡Boston University and Tel Aviv University. Email: canetti@bu.edu. Supported by the Check Point Institute for Information Security, an NSF EAGER grant, and NSF Algorithmic foundations grant 1218461.

§Boston University. Email: omer@bu.edu. Supported by the Simons award for graduate students in theoretical computer science and an NSF Algorithmic foundations grant 1218461.

¶Efi Arazi School of Computer Science, IDC Herzliya, Israel. Email: alon.rosen@idc.ac.il. 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.

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 without knowing 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 outputs 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 1.1 (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 1.2 (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 [KRR14], based, for instance, on the Learning with Errors Assumption. We additionally show that the limitation given by Theorem 1.1 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 1.3 (informal)**.**

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

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

[1] See discussion in [CP13] on the equivalent concept of 2-message P-certificates.

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, SW14, 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, BCC$^+$14]. 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 [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 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

[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.

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 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 [KRR14] construct a privately verifiable P-delegation scheme based on any private information retrieval scheme with sub-exponential security. While the scheme of [KRR14] 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)see Appendix A for

more details). 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. In Section 5.4, we present the zero-knowledge protocols constructed from GEOWFs. In Section A, we discuss relevant black-box lower for EOWFs, and ZK.

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

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

associated with an efficient (probabilistic) key sampler $\mathcal{K}_{\mathcal{F}}$, is 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 $z \in \{0,1\}^{b(n)}$:*

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

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 $z \in \{0,1\}^{b(n)}$:*

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

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 2.2 (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*

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

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

1. **One-way:** *As in Definition 2.1.*

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 $z \in \{0,1\}^{b(n)}$:*

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

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 2.1 (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 b-bounded auxiliary-input verifiers (see Section 5),as well as for any application where testing if the adversary breaks the scheme can be done efficiently.

Remark 2.2 (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 $\mathcal{R}_e^{\mathcal{F}}(f_e(x), x')$ 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 if the function is keyed). 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 2.1, 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 2.3 (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 $\mathcal{R}_e^{\mathcal{F}}(y, x)$ 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 $z \in \{0,1\}^{b(n)}$:*

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

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 $z \in \{0,1\}^{b(n)}$:*

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

We further say that the function is

- **Publicly-verifiable** *if $\mathcal{R}_e^{\mathcal{F}}(f_e(x), x')$ can always be efficiently computed by a tester $\mathcal{T}(e, f_e(x), x')$.*
- **Privately-verifiable** *if $\mathcal{R}_e^{\mathcal{F}}(f_e(x), x')$ 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 2.3, whereas $\mathcal{R}^{\mathcal{F}}$-hardness is only against adversaries with auxiliary input of an apriori fixed polynomial size $b(n)$.

Remark 2.3 (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 $\mathcal{R}^{\mathcal{F}}$ includes all pairs $(f_e(x), x)$, and thus ensure that $\mathcal{R}^{\mathcal{F}}$-hardness does imply one-wayness.

Remark 2.4 (GEOWFs vs. 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 $f_e(x)$ is 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. GEOWF 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 $f_e(x) \sim f_e(x')$. In particular, the limitations we show in Section 3 on GEOWFs apply to proximity EOWFs as well.

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. We start by defining $i\mathcal{O}$ and puncturable PRFs.

3.1 Indistinguishability Obfuscation

Indistinguishability obfuscation was introduced in [BGI$^+$01] and given a candidate construction in [GGH$^+$13b], and subsequently in [BR13, BGTK$^+$13].

Definition 3.1 (Indistinguishability obfuscation [BGI$^+$01])**.** *A PPT algorithm $i\mathcal{O}$ is said to be an* indistinguishability obfuscator *(INDO) for $\mathcal{C}$, if it satisfies:*

1. **Functionality:** *For any $C \in \mathcal{C}$,*

$$\Pr_{i\mathcal{O}}[\forall x : i\mathcal{O}(C)(x) = C(x)] = 1 \ .$$

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 in each pair are of the same size and functionality, it holds that:*

$$\left\{i\mathcal{O}(C_n^{(1)})\right\}_{n\in\mathbb{N}} \approx_c \left\{i\mathcal{O}(C_n^{(2)})\right\}_{n\in\mathbb{N}} \ .$$

Remark 3.1 (Efficiently-falsifiable $i\mathcal{O}$). The assumption that $i\mathcal{O}$ obfuscators exist according to the above (standard) formulation is not *efficiently falsifiable* in the language of [Nao03]. Specifically, it can be formulated as a *cryptographic game* [DOP05, HH09] between a challenger and an attacker: the attacker submits two circuits $\mathcal{C}_0, \mathcal{C}_1$, gets an obfuscation $i\mathcal{O}(\mathcal{C}_b)$ for a random b, and has to guess b. However, the challenger cannot efficiently check whether the adversarially chosen circuits indeed compute the same function. Thus, it is not efficiently falsifiable.

We note, however, that for our specific application, we can settle for a restriction of $i\mathcal{O}$ that is falsifiable: we only require that, for some fixed efficiently samplable distribution $\mathcal{D}$ on pairs of circuits with the same functionality, $i\mathcal{O}$ holds for an honestly sampled pair. That is, instead of letting the attacker adversarially choose the circuits, the challenger samples a pair of circuits $\mathcal{C}_0, \mathcal{C}_1$ from $\mathcal{D}$ and hands them to attacker together with $i\mathcal{O}(\mathcal{C}_b)$.

3.2 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].

Definition 3.2 (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 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 $x^* \in \{0,1\}^{\ell(n)}$,*

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

2. **Indistinguishability at punctured points:** *The following ensembles are computationally indistinguishable:*

   - $\{x^*, k_{x^*}, \mathsf{PRF}_k(x^*) \mid k \leftarrow \mathcal{K}_{\mathcal{PRF}}(1^n), k_{x^*} = \mathsf{Punc}(k, x^*)\}_{x^*\in\{0,1\}^{m(n)}, n\in\mathbb{N}}$
   - $\left\{x^*, k_{x^*}, u \;\middle|\; k \leftarrow \mathcal{K}_{\mathcal{PRF}}(1^n), k_{x^*} = \mathsf{Punc}(k, x^*), u \leftarrow \{0,1\}^{\ell(n)}\right\}_{x^*\in\{0,1\}^{m(n)}, n\in\mathbb{N}}$.

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. As shown in [BGI13, BW13, KPTZ13], the GGM [GGM86] PRF yield puncturable PRFs as defined above.

3.3 The Impossibility Result

We now show that if indistinguishability obfuscators exist, there do not exist auxiliary-input EOWFs or generalized EOWFs (GEOWFs) according to Definitions 2.1,2.3:

Theorem 3.1. *Assuming indistinguishability obfuscation for all circuits, neither EOWFs nor GEOWFs exist, with respect to common auxiliary-input of unbounded polynomial length.*

Before proving the theorem two remarks are in place:

Remark 3.2 (Implications for 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 proximity EOWFs, which as noted in Remark 2.4 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 3.3 (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 if we only require extractability with respect to *individual* auxiliary-input, as defined in Remark 2.2. 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 2.2), such as the uniform distribution.

To prove the Theorem 3.1, 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, for 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.

3.3.1 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 $z(e)$.

3.3.2 The Auxiliary Input Distribution

Let $\mathcal{F}$ be a family of extractable one-way functions and let $\mathcal{PRF}$ be a puncturable pseudo-random function family. We start by defining two families of circuits

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

The 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.

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 $y = f_e(x)$.

Figure 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^*.

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

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.

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.

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.

Figure 3: The auxiliary input distribution Z_n.

3.3.3 $\mathcal{A}$ Does Not Have an Extractor

We next show that $\mathcal{A}$ cannot have any extractor $\mathcal{E}$ satisfying Definition 2.1. In fact, we show a stronger claim; namely, that for the auxiliary input distribution $\mathcal{Z}$, any extractor fails with overwhelming probability.

Proposition 3.1. *Let $\mathcal{E}$ be any PPT candidate extractor for $\mathcal{A}$ then*

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

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

Proof of Proposition 3.1. 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 towards 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, $z=i\mathcal{O}([C_k]_s)$.

Next, 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 fact that, for any e^* and k, C_k and $C_{k_{e^*},y^*}$ compute the same function, and the $i\mathcal{O}$ indistinguishability guarantee.

Next, we consider another experiment where $Z_n(e^*,y^*)$ is altered to a new distribution $Z_n(e^*,u)$ 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 pseudo-random, even given the punctured key k_{e^*}.

This means 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 generalized EOWF (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 pre-image x of $y = z(e)$, we start from an extractor that finds $x \in \mathcal{R}^{\mathcal{F}}_e(y)$. 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 proofs 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 pseudo-randomness at punctured points, in the second, it can indeed be efficiently tested whether the extractor successfully obtained a witness $x \in \mathcal{R}^{\mathcal{F}}_e(y)$. This is clear in the case that $\mathcal{R}^{\mathcal{F}}$ is publicly-verifiable and 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 3.1, we note that puncturable PRFs can be constructed from one-way functions. Furthermore, EOWF is already a OWF, and any GEOWF with $\mathcal{R}^{\mathcal{F}}$-hardness implies that NP $\neq$ coRP, which in conjunction with $i\mathcal{O}$ implies OWFs [MR13]. Thus, the impossibility of auxiliary-input EOWFs and GEOWFs is implied by indistinguishability obfuscation without any further assumptions.

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 [KRR14].

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 $T(n) \in (2^{\omega(\log n)}, 2^{\mathrm{poly}(n)})$ 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 4.1 (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 $S \subseteq \{0,1\}^{\mathrm{poly}(n)} \setminus \mathcal{L}_{\mathcal{U}}(T(n))$:*

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

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

Existence and connection to delegation of computation. There are two differences between the notion of delegation for deterministic computations (See, e.g., [KRR14]) and the NIUA notion defined above. The first is that 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 $T_{\mathcal{M}}$. The second difference is that only non-adaptive 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 Dtime$(T) \subseteq$ EXP, assuming subexponentially secure private information retrieval schemes, which can in turn be constructed based the subexponential Learning with Errors assumption [BV11].

In order to get a (privately verifiable) NIUA for 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 $\mathrm{poly}(T)$, 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 super-polynomial. Fortunately, a rather standard transformation does allow 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 for all $t \in \left\{1, 2, 2^2, \ldots, 2^{\log T}\right\}$, 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 $\mathrm{poly}(2t_{\mathcal{M}})$ as required.

As for adaptivity, in their scheme, the generator does work independently of the input x, but only non-adaptive soundness is shown; namely, soundness is only guaranteed 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 $O(b(n))$ 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 (see e.g., [BIN97]). Then, we can take a union bound over all $2^{b(n)}$ adaptive choices of x and get 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 4.1 (Following from [KRR14])**.** *Assuming the Learning with Errors Problem is sub-exponentially hard, for any $b(n) = \mathrm{poly}(n)$, and $T(n) \in (2^{\omega(\log n)}, 2^{\mathrm{poly}(n)})$, 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 $T(n) \in (2^{\omega(\log n)}, 2^{\mathrm{poly}(n)})$, 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 $f_n : \{0,1\}^{\ell(n)} \to \{0,1\}^{\ell'(n)}$, 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 $\mathcal{R}^{\mathcal{F}}$.

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 4: 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 5, 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).

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

Claim 4.1. *$\mathcal{R}^{\mathcal{F}}$ 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 $\mathcal{R}^{\mathcal{F}}_n$ can be efficiently tested.

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

Remark 4.1 (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 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 4.2. *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 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 $\mathcal{M}_z$.

Proof of Theorem 4.2. 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 NIUA (for $2^{b(n)}$ statements), with probability $\varepsilon(n) - \mathrm{negl}(n)$. $\mathcal{P}^*$, 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 $\varepsilon(n) - \mathrm{negl}(n)$, 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 pseudo-random 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 NIUA.

$\mathcal{R}^{\mathcal{F}}$-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 and PPT adversary $\mathcal{M}$ with advice of size at most $b(n) - \omega(1)$. For an 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 runs $\mathcal{M}(1^n; z)$. The extractor $\mathcal{E}$ is given $(\mathcal{M}, z)$, where $\mathcal{M}_z$ has description size at most $b(n)$ and running time at most $t_{\mathcal{M}} < T(n)$, and $\mathcal{M}_z(1^n) = y = (\sigma, v) \in \mathsf{Image}(f_n)$. To compute a witness $x' \in \mathcal{R}^{\mathcal{F}}(y)$, $\mathcal{E}$ computes a certificate π for the fact that "$\mathcal{M}_z(1^n) = v$", and then outputs $x' = (\mathcal{M}_z, \pi, \mathsf{pad})$. The fact that x' is indeed a valid witness follows directly from the perfect completeness of the scheme. Finally, we note that by the relative prover efficiency of the NIUA the extractor runs in time that is polynomial in the running time $t_{\mathcal{M}}$ of the adversary . □

Remark 4.2 ($\mathcal{R}^{\mathcal{F}}$-hardness against superpolynomial adversaries). In Section 5.4.2, 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 an NIUA that is sound against such adversaries (such an NIUA can be obtained from [KRR14], based on an appropriately strong private information retrieval scheme).

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 4.2 (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.

[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.

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 6.

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 6: The function f_n.

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

Theorem 4.3. *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 NIUA as in the proof of Theorem 4.2 (leveraging the fact that a truly random $\tilde{v}$ almost surely cannot be computed by a short machine).

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 $i = 1^n$.

Proof of Theorem 4.3. We first show $\mathcal{R}^{\mathcal{F}}$-hardness, and then show $\mathcal{R}^{\mathcal{F}}$-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 NIUA (for $2^{b(n)}$ statements), with probability $\varepsilon(n) - \mathrm{negl}(n)$. $\mathcal{P}^*$ is defined as in the proof of Theorem 4.2: 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 4.2. *With probability $\varepsilon(n) - \mathrm{negl}(n)$, 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 since, as in the proof of Theorem 4.2, 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 NIUA.

Proof. To prove the claim, we first consider an 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 $\mathrm{negl}(n)$. 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 and PPT adversary $\mathcal{M}$ with advice of size at most $b(n) - \omega(1)$. The proof is similar to the extractability proof of Theorem 4.2. For an adversarial code $\mathcal{M}$ and advice $z \in \{0,1\}^{b(n)-\omega(1)}$, we denote by $\mathcal{M}_z$ the machine that, on input 1^n, runs runs $\mathcal{M}(1^n; z)$. The extractor $\mathcal{E}$ is given $(\mathcal{M}, z)$, where $\mathcal{M}_z$ has description size at most $b(n)$ and running time at most $t_{\mathcal{M}} < T(n)$, and $\mathcal{M}_z(1^n) = (\sigma, v) \in \mathsf{Image}(f_n)$.

If $(\sigma, v) \neq (0^{m(n)}, 0^{b(n)+n})$, we know that σ must be valid, in which case $\mathcal{E}$ computes a certificate π for the fact that "$\mathcal{M}_z(1^n) = v$", and then outputs the preimage $x' = (0^n, (0^n, 0^n), (\sigma, \mathcal{M}_z, v, \pi))$. The fact that x' is indeed a valid preimage follows directly from the perfect completeness of the scheme, for a valid σ. If $(\sigma, v) = (0^{m(n)}, 0^{b(n)+n})$, the extractor outputs the preimage $x' = (1^n, (0^n, 0^n), (0^{m(n)}, 0^{b(n)}, 0^{b(n)+n}, 0^{p(n)}))$.

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

5 2-Message and 3-Message Zero Knowledge against Bounded-Auxiliary-Input Verifiers

In this section, we define and construct two and three 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.

5.1 Definition

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

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

Definition 5.1. *An argument system (P, V) for an NP relation $\mathcal{R}_{\mathcal{L}}(\varphi, w)$ is zero knowledge against verifiers with b-bounded advice if for every PPT verifier V^*, there exists a PPT simulator $\mathcal{S}$ such that:*

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

where computational indistinguishability is with respect to arbitrary non-uniform distinguishers.

Remark 5.1 (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 does not hold since t_{V^*} is unbounded and can be larger than the bound b on the size of the advice.

5.2 WI Proof of Knowledge with an Instance-Independent First Message

In this section, we define and construct 3-message WI proofs of knowledge with an instance-independent first message, which will be used 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 length in a security parameter n, independently of $|\varphi, w|$.

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 do satisfy the independence requirement (as there is no first prover message at all), but they do not constitute a proof of knowledge.

We show that, using 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 only depends on $|\varphi|$.

Definition 5.2 (WIPOK with instance-independent first message)**.** *Let $\langle P \leftrightarrows V \rangle$ be a 3-message proof system for $\mathcal{L}$ with messages (α, β, γ); we say it is a WIPOK with instance-independent first message, if it satisfies:*

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

$$\Pr\left[V(\varphi, \alpha, \beta, \gamma; r') = 1 \;\middle|\; \begin{array}{r} \alpha \leftarrow P(1^n; r) \\ \beta \leftarrow V(\ell, \alpha; r') \\ \gamma \leftarrow P(\varphi, w, \alpha, \beta; r) \end{array} \right] = 1 \ ,$$

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

   The honest prover's first message α is of length n, independently of the length of the statement and witness (φ, w).

2. **Adaptive witness-indistinguishability:** *for any deterministic polysize verifier V^* and all large enough $n \in \mathbb{N}$:*

$$\Pr\left[V^*(\varphi, \alpha, \beta, \gamma) = b \;\middle|\; \begin{array}{r} \alpha \leftarrow P(1^n; r) \\ \varphi, w_0, w_1, \beta \leftarrow V^*(\alpha) \\ \gamma \leftarrow P(\varphi, w_b, \alpha, \beta; r) \end{array} \right] \leq \frac{1}{2} + \mathrm{negl}(n) \ ,$$

   where $b \leftarrow \{0,1\}$, $r \leftarrow \{0,1\}^{\mathrm{poly}(n)}$ is the randomness used by P, and $w_0, w_1 \in \mathcal{R}_{\mathcal{L}}(\varphi)$.

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

$$\textit{if } \Pr\left[V(\varphi, \alpha, \beta, \gamma; r') = 1 \;\middle|\; \begin{array}{r} \alpha \leftarrow P^* \\ \beta \leftarrow V(\ell(n), \alpha; r') \\ \varphi, \gamma \leftarrow P^*(\alpha, \beta) \end{array} \right] \geq \varepsilon \ ,$$

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

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

Construction from ZAPs. 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 do not provide POK. The high-level idea is to add the POK 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 (e.g., as formalized in [PW09]), 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 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, BP13a].

In what follows, we denote by $(\mathcal{C}, \mathcal{R})$ the committer and receiver algorithms of a perfectly-binding 3-message extractable commitment protocol, and we denote by $\vec{C} = \left(C^{(1)}, C^{(2)}, C^{(3)}\right)$ its three messages. We further require that extraction is possible given any two valid transcripts $\vec{C}, \vec{C}'$ that share the same first message. Such an extractable commitment can be constructed from any perfectly-binding non-interactive commitment, see e.g. [PW09].

Protocol 7

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

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 $(C_0^{(1)}, C_1^{(1)}, C^{(1)}) \leftarrow (\mathcal{C}(s_0), \mathcal{C}(s_1), \mathcal{C}(b))$, where $|(C_0^{(1)}, C_1^{(1)}, C^{(1)})| = n$.[a]

2. V, given the length of the statement $\ell = |\varphi|$, samples randomness $r \leftarrow \{0,1\}^{\mathrm{poly}(n)}$ 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), now performs the following:

   - computes the third committer messages $(C_0^{(3)}, C_1^{(3)}, C^{(3)}) \leftarrow (\mathcal{C}(s_0, C_0^{(2)}), \mathcal{C}(s_1, C_1^{(2)}), \mathcal{C}(b, C^{(2)}))$.
   - 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:

$$\left\{ \left\{ \vec{C} = \mathcal{C}(0, C^{(2)}) \right\} \vee \left\{ \begin{array}{c} \vec{C}_0 = \mathcal{C}(s_0, C_0^{(2)}) \\ a_0 = w \oplus \mathsf{PRG}(s_0) \\ w \in \mathcal{R}_{\mathcal{L}}(\varphi) \end{array} \right\} \right\} \bigwedge$$
$$\left\{ \left\{ \vec{C} = \mathcal{C}(1, C^{(2)}) \right\} \vee \left\{ \begin{array}{c} \vec{C}_1 = \mathcal{C}(s_1, C_1^{(2)}) \\ a_1 = w \oplus \mathsf{PRG}(s_1) \\ w \in \mathcal{R}_{\mathcal{L}}(\varphi) \end{array} \right\} \right\}$$

   - 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.

Figure 7: A 3-message WIPOK with instance-independent first message

Lemma 5.1. *Protocol 7 is a 3-message WIPOK with instance-independent first message.*

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

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.

We next prove the lemma. The proof is an adaptation of a proof from [BP13a].

Proof. We start by showing 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 next consider a sequence of hybrids starting with an hybrid describing an interaction with a prover that uses $w_0 \in \bar{w}_0$, and ending with an 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 list of hybrids is given 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, the ZAP uses the witness $((s_0, w_0), (s_1, w_0))$; 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. In Table 1, the witness used in part 0 of the ZAP is referred to as zapw_0, and the one corresponding to 1 in 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 $((s_b, w_0), (s_{1-b}, w_0))$. (Again the witness should include the randomness for the commitment $\vec{C}$, and $\vec{C}_{1-b}$, but is omitted from our notation.) 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 $u \leftarrow \{0,1\}^{|\mathsf{PRG}(s_b)|}$. Computational indistinguishability of this hybrid and the previous one, follows pseudorandomness.

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 backwards).

Hybrids 1.1 *to* 1.6*:* These hybrids are symmetric to the above hybrids, only that they start from w_1 instead of w_0. This means that they end in 1.6 which uses an 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 towards 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 WLOG 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 that it locates w_0 and w_1 according to 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.

It is left to 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 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 Lemma 5.1. □

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 three message definition the verifier message does not depend on the instance, but is allowed to depend on its length. In such a protocol, we only require 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 instance-independent first message.)

5.3 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 5.3. *A scheme* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Eval}, \mathsf{Dec})$*, where* $\mathsf{Gen}, \mathsf{Eval}$ *are probabilistic and* $\mathsf{Enc}, \mathsf{Dec}$ *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:*

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

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

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

- **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 $\mathcal{S}$ such that:*

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

5.4 Constructions

In this section, we construct zero-knowledge protocols against verifiers with bounded advice from generalized extractable one-way functions 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 (non-interactive) commitments that can be inverted in super-poly time $T(n)$, GEOWFs that are hard against $\mathrm{poly}(T(n))$-size adversaries, and any 2-message WI with instance-independent verifier message (in particular, ZAPs).

5.4.1 A 3-message zero-knowledge argument of knowledge

Let $(\mathsf{Gen},\mathsf{Enc},\mathsf{Eval},\mathsf{Dec})$ be a semantically-secure, circuit-private, 1-hop homomorphic encryption scheme. Let $(\mathsf{wi}_1,\mathsf{wi}_2,\mathsf{wi}_3)$ denote the messages of 3-message WIPOK with an instance-independent first message (as in Definition 5.2). Let $\mathcal{F} = \{f_n\}_{n\in\mathbb{N}}$ be a key-less GEOWF, against $(b(n)+2n)$-bounded-auxiliary-input adversaries, with respect to a privately-verifiable relation $\mathcal{R}^{\mathcal{F}} = \left\{\mathcal{R}_n^{\mathcal{F}}\right\}_{n\in\mathbb{N}}$. Let $\mathcal{T}(x,x')$ be the efficient tester for $\mathcal{R}_n^{\mathcal{F}}(f_n(x),x')$. We further 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 $\mathcal{R}_n^{\mathcal{F}}(f_n(x),x')=1$". Also, let $\mathbf{1}$ be a circuit of the same size as $\mathcal{T}_{y,x'}$ that always returns 1. The protocol is given in Figure 8.

Theorem 5.1. *Protocol 8 is a zero-knowledge 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

Protocol 8

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

Auxiliary Input to P: a witness w for φ.

1. P sends the first message $\mathsf{wi}_1 \in \{0,1\}^n$ of the instance-dependent WIPOK.
2. V samples $x \leftarrow \{0,1\}^{\ell(n)}$ and $\mathsf{sk} \leftarrow \mathsf{Gen}(1^n)$, computes $y = f_n(x)$, $\mathsf{c}_x = \mathsf{Enc}_{\mathsf{sk}}(x)$ and sends (y, c_x) , as well as the second WIPOK message wi_2.
3. P samples $\hat{\mathsf{c}} \leftarrow \mathsf{Eval}(\mathbf{1}, \mathsf{c}_x)$, and sends $\hat{\mathsf{c}}$, together with the WIPOK message wi_3 stating that:

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

   using the witness $w \in \mathcal{R}_{\mathcal{L}}(\varphi)$.
4. V verifies the proof and that $\mathsf{Dec}_{\mathsf{sk}}(\hat{\mathsf{c}}) = 1$.

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

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 together with its encrypted preimage x, means that x' must be such that x' satisfies $\mathcal{R}^{\mathcal{F}}(f_n(x), x') = 1$, and thus the prover actually 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 use the fact that if the verifier sends y together with an encryption of a true preimage x, the 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 auxiliary-input is appropriately bounded. In case, the verifier diverges from the protocol, and doesn't send proper y and encrypted preimage, the definition of $\mathcal{T}_{y,x'}$ guarantees that the circuit will also accept in this case. 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.

Claim 5.1. *Protocol 8 is an argument of knowledge against against arbitrary polysize provers.*

Proof. Let P^* be any polysize prover that convinces V of accepting with noticeable probability $\varepsilon(n)$. The witness extractor would derive from P^* a new prover for P^*_{wi} that emulates P^* in the WIPOK; in particular, it would honestly sample (y, c_x) as part of the second verifier message that P^* gets. The extractor would then choose the random coins r for P^*_{wi}, sample a transcript tr of an execution with the honest WIPOK verifier V_{wi}, and apply the WIPOK extractor on the transcript tr, with oracle access to P^*_{wi}. The WIPOK extractor then hopefully obtains a witness for the WI statement

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

where (y, c_x) are those honestly sampled by P^*_{wi}, and $\hat{\mathsf{c}}$ is output by P^*.

We claim that, with noticeable probability $\varepsilon(n)^2/2 - \mathrm{negl}(n)$, we find a witness w for the first part of the statement $\varphi \in \mathcal{L}$. Otherwise, we can use P^* to break the $\mathcal{R}^{\mathcal{F}}$-hardness of $\mathcal{F}$. To prove this claim, we first note that the emulated transcript tr in this experiment is distributed identically to the transcript in a real execution of P^* with the honest verifier. Thus, we know that such a transcript tr is accepted by V with probability at least $\varepsilon(n)$. Now, let us call random coins r for P^*_{wi} good if they are such that with probability at least $\varepsilon(n)/2$ over the coins of the WIPOK verifier V_{wi}, it accepts the proof given by P^*_{wi}. Since we know that overall V_{wi} accepts with probability at least $\varepsilon(n)$, then by a standard averaging argument, at least an $\varepsilon(n)/2$ fraction of the coins r for P^*_{wi} are good. Furthermore, conditioned on a transcript tr that is accepted by V, the probability that the corresponding coins r are good increases. Thus, it follows that the probability that tr is accepting and the corresponding coins r are good is at least $\varepsilon(n) \cdot \varepsilon(n)/2$. Now, recall that, whenever this occurs, the extractor for the WIPOK would also output a witness for the corresponding statement (except with negligible probability).

We would like to show that the extracted witness is the one for the $\varphi \in \mathcal{L}$ statement. Indeed, assume that, with noticeable probability $\eta(n)$, it holds that tr is accepting, the extractor outputs a witness, but the witness is for the second statement. This, in particular, means that the witness extractor outputs x' such that $\hat{\mathsf{c}} = \mathsf{Eval}(\mathcal{T}_{y,x'}, \mathsf{c}_x)$, where $\hat{\mathsf{c}}$ is the output of P^*. Moreover, since the transcript is accepted by V, we know that $\mathsf{Dec}(\hat{\mathsf{c}}) = 1$. By correctness of decryption, this means that $\mathcal{T}_{y,x'}(x) = 1$, which in turn implies that $\mathcal{T}(x, x') = 1$, since $y = f_n(x)$. In other words, x' is a valid witness satisfying $\mathcal{R}^{\mathcal{F}}_n(f_n(x), x') = 1$.

We can now construct a breaker for the $\mathcal{R}^{\mathcal{F}}$-hardness of $\mathcal{F}$. The breaker, given $y = f_n(x)$, would simply emulate all of the experiment above on its own, where P_{wi} would use y, and an encryption of zero $\mathsf{c}_0 = \mathsf{Enc}_{\mathsf{sk}}(0)$ to emulate the second verifier message, instead of sampling $(y = f_n(x), \mathsf{c}_x)$ on its own. We claim that it would obtain the desired witness x' with noticeable probability $\eta(n) - \mathrm{negl}(n)$. Indeed, had we used an encryption c_x of the preimage of y, instead of a zero-encryption, we know that it would produce a valid witness x with probability $\eta(n)$. Thus, the claim follows by the semantic security of Enc. This completes the proof of Claim 5.1 □

We next show that the protocol is ZK. We note that, since the ZK simulator is allowed to simulate the (apriori unbounded) randomness of the verifier V^*, we can restrict attention to verifiers V^* that only have bounded randomness. Indeed (assuming there exist OWFs), we can always consider a new verifier $\widetilde{V}^*$ that first stretches its bounded randomness using a PRG and then emulates V^*. Then 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^*. In particular, from hereon we we can simply focus on deterministic verifiers V^* that get their bounded randomness as part of their bounded advice.

Claim 5.2. *Protocol 8 is ZK against any polytime verifier V^* with auxiliary-input of size at most $b(n)$.*

Proof. We describe a universal ZK simulator $\mathcal{S}$ and show its validity (universality is in the sense of Remark 5.1). Let $\varphi \in \mathcal{L}$ and 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 honestly computing the first message $\mathsf{wi}_1 \in \{0,1\}^n$ of the WIPOK with instance-independent first message. It then 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.

$\mathcal{S}$ now 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^*. 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 $\mathrm{poly}(t^*_V)$. $\mathcal{S}$ does not test whether y is a valid image directly,

it applies the extractor regardless to obtain a candidate x', and then computes $\hat{\mathsf{c}} = \mathsf{Eval}(\mathcal{T}_{y,x'}, \mathsf{c})$. Then it sends $\hat{\mathsf{c}}$ to V^*, and completes the WIPOK using the trapdoor x' as a witness.

The validity of the simulator now follows by witness indistinguishability, as well as by the circuit privacy guarantee given by Eval. Specifically, we first move to a hybrid simulator $\mathcal{S}'$ that proves the WIPOK statement using the actual witness w. The view generated by $\mathcal{S}'$ is indistinguishable from the one generated by $\mathcal{S}$ due to the WI property.

Now, we claim that the view generated by $\mathcal{S}'$ is indistinguishable from that generated by honest prover P. First, note that the only difference between the two is that P sends $\hat{\mathsf{c}} \leftarrow \mathsf{Eval}(\mathbf{1}, \mathsf{c})$, whereas $\mathcal{S}'$ sends $\hat{\mathsf{c}} \leftarrow \mathsf{Eval}(\mathcal{T}_{y,x'}, \mathsf{c})$, for the extracted input x'. Now, note that if c is a valid ciphertext, then $\mathcal{T}_{y,x'}(\mathsf{Dec}(\mathsf{c})) = \mathbf{1}(\mathsf{Dec}(\mathsf{c})) = 1$; indeed, if $y = f_n(x)$ where $x = \mathsf{Dec}(\mathsf{c})$, then the extracted x' is such that $\mathcal{T}(x, x') = 1$, and the above follows by the definition of $\mathcal{T}_{y,x'}(x)$. Thus, in this case, the distribution of $\hat{\mathsf{c}}$ induced by P is indistinguishable from that induced by $\mathcal{S}'$, by circuit privacy. In fact, circuit privacy says that this is also the case if c is an invalid cipher. □

This completes the proof of Theorem 5.1. □

5.4.2 A 2-message zero-knowledge 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.

Let $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Eval}, \mathsf{Dec})$ be a semantically-secure, circuit-private, 1-hop homomorphic encryption scheme. Let $(\mathsf{wi}_1, \mathsf{wi}_2)$ denote the messages of 2-message WI with an instance-independent first message (as in Definition 5.2). Let $\mathcal{F} = \{f_n\}_{n\in\mathbb{N}}$ be a key-less GEOWF, against $(b(n) + n)$-bounded-auxiliary-input adversaries, with respect to a privately-verifiable relation $\mathcal{R}^{\mathcal{F}} = \left\{\mathcal{R}_n^{\mathcal{F}}\right\}_{n\in\mathbb{N}}$. Further assume that $\mathcal{F}$ is one-way against adversaries of size $\mathrm{poly}(T)$ (see Remark 4.2). Let $\mathcal{T}(x, x')$ be the efficient tester for $\mathcal{R}_n^{\mathcal{F}}(f_n(x), x')$. We further 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 $\mathcal{R}_n^{\mathcal{F}}(f_n(x), x') = 1$". Also, let $\mathbf{1}$ be a circuit of the same size as $\mathcal{T}_{y,x'}$ that always returns 1. Let $\mathcal{C}$ be a perfectly binding commitment that is hiding against polysize adversaries, and can be completely inverted in time $T(n)$, for some computable super-polynomial function $T(n) = n^{\omega(1)}$. The protocol is given in Figure 9.

Theorem 5.2. *Protocol 9 is a zero-knowledge 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 that now the simulator also commits to the input that it extracts from the verifier (and by the hiding of the commitment ZK is maintained). The proof of soundness is essentially the same as showing POK in the 3-message protocol, only that now, the WI 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.

A more detailed proof follows.

Proof sketch. We first show that the protocol is a sound against polysize adversaries.

Claim 5.3. *Protocol 9 is an argument.*

Proof sketch. Let P^* be any polysize prover, and assume towards contradiction that for infinitely many $\varphi \notin \mathcal{L}$, P^* convinces V of accepting with noticeable probability $\varepsilon(n)$. We show to break the $\mathcal{R}^{\mathcal{F}}$-hardness

Protocol 9

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

Auxiliary Input to P: a witness w for φ.

1. V samples $x \leftarrow \{0,1\}^{\ell(n)}$ and $\mathsf{sk} \leftarrow \mathsf{Gen}(1^n)$, computes $y = f_n(x)$, $\mathsf{c}_x = \mathsf{Enc}_{\mathsf{sk}}(x)$ and sends (y, c_x) , as well as the first WI message wi_1.
2. P samples a commitment to zero $C \leftarrow \mathcal{C}(0^\ell)$, $\hat{\mathsf{c}} \leftarrow \mathsf{Eval}(\mathbf{1}, \mathsf{c}_x)$, and sends $(C, \hat{\mathsf{c}})$, together with the second WI message wi_2 stating that:

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

   using the witness $w \in \mathcal{R}_{\mathcal{L}}(\varphi)$.
3. V verifies the proof and that $\mathsf{Dec}_{\mathsf{sk}}(\hat{\mathsf{c}}) = 1$.

Figure 9: A 2-message ZK argument against verifiers with b-bounded auxiliary input.

of $\mathcal{F}$. The breaker, given y would sample a first WI message wi_1, and encryption of zero c_0, and feed $(y, \mathsf{c}_0, \mathsf{wi}_1)$ to P^*, who outputs a commitment C, an alleged image y, and a proof wi_2 for the statement

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

By the semantic security of the 1-hop encryption, the above is indistinguishable from an experiment in which the breaker uses c_x for an actual preimage of y, and thus we know that with probability $\varepsilon(n) - \mathrm{negl}(n$ the proof is convincing. By the soundness of the WI system, and since $\varphi \notin \mathcal{L}$, it follows that C is a commitment to a proper witness x'. The inverter can now break C in time $T(n)$ and thus break $\mathcal{R}^{\mathcal{F}}$-hardness of $\mathcal{F}$. □

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 5.4. *Protocol 9 is ZK against any polytime verifier V^* with advice of size at most $b(n)$.*

Proof sketch. We describe a universal ZK simulator $\mathcal{S}$ and show its validity (universality is in the sense of Remark 5.1). Let $\varphi \in \mathcal{L}$ and 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 $(y, \mathsf{c}, \mathsf{wi}_1)$ that are (allegedly) an image of the of the function f_n, an encryption of its preimage, and the verifier message of the WI protocol.

$\mathcal{S}$ 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 $\mathrm{poly}(t^*_V)$.

$\mathcal{S}$ now computes $\hat{\mathsf{c}} = \mathsf{Eval}(\mathcal{T}_{y,x'}, \mathsf{c})$, as well as a commitment C to x', and completes the WI using the trapdoor x' as a witness. 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 simulator $\mathcal{S}'$ that proves the WI statement using the witness w. The view generated by $\mathcal{S}'$ is indistinguishable from the one generated by $\mathcal{S}$ due to the WI property. Now, we can claim that the view generated by $\mathcal{S}'$ is indistinguishable from that generated by the honest prover P. Indeed, the only difference between the two is that P commits to 0^ℓ instead of x', and sends $\hat{\mathsf{c}} \leftarrow \mathsf{Eval}(\mathbf{1}, \mathsf{c})$, whereas $\mathcal{S}'$ sends $\hat{\mathsf{c}} \leftarrow \mathsf{Eval}(\mathcal{T}_{y,x'}, \mathsf{c})$, 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 5.2). □

This completes the proof of Theorem 5.2. □

Acknowledgements

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.

References

[AIR01] William Aiello, Yuval Ishai, and Omer Reingold. Priced oblivious transfer: How to sell digital goods. In *EUROCRYPT*, pages 119–135, 2001.

[Bar01] Boaz Barak. How to go beyond the black-box simulation barrier. In *FOCS*, pages 106–115, 2001.

[BC12] Nir Bitansky and Alessandro Chiesa. Succinct arguments from multi-prover interactive proofs and their efficiency benefits. In *CRYPTO*, pages 255–272, 2012.

[BCC+13] Nir Bitansky, Ran Canetti, Alessandro Chiesa, Shafi Goldwasser, Huijia Lin, Eran Tromer, and Aviad Rubinstein. The haunting of the snark. *Manuscript*, 2013.

[BCC+14] Nir Bitansky, Ran Canetti, Henry Cohn, Shafi Goldwasser, Yael Tauman Kalai, Omer Paneth, and Alon Rosen. The impossibility of obfuscation with auxiliary input or a universal simulator. *CoRR*, abs/1401.0348, 2014.

[BCCT12] Nir Bitansky, Ran Canetti, Alessandro Chiesa, and Eran 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, pages 326–349, 2012.

[BCCT13] Nir Bitansky, Ran Canetti, Alessandro Chiesa, and Eran Tromer. Recursive composition and bootstrapping for snarks and proof-carrying data. In *STOC*, pages 111–120, 2013.

[BCI+13] Nir Bitansky, Alessandro Chiesa, Yuval Ishai, Rafail Ostrovsky, and Omer Paneth. Succinct non-interactive arguments via linear interactive proofs. In *TCC*, pages 315–333, 2013.

[BGI+01] Boaz Barak, Oded Goldreich, Russell Impagliazzo, Steven Rudich, Amit Sahai, Salil P. Vadhan, and Ke Yang. On the (im)possibility of obfuscating programs. In *CRYPTO*, pages 1–18, 2001.

[BGI13] Elette Boyle, Shafi Goldwasser, and Ioana Ivan. Functional signatures and pseudorandom functions. *IACR Cryptology ePrint Archive*, 2013:401, 2013.

[BGTK+13] Boaz Barak, Sanjam Garg, Yael Tauman-Kalai, Omer Paneth, and Amit Sahai. Protecting obfuscation against algebraic attacks. *IACR Cryptology ePrint Archive*, 2013:631, 2013.

[BIN97] Mihir Bellare, Russell Impagliazzo, and Moni Naor. Does parallel repetition lower the error in computationally sound protocols? In *FOCS*, pages 374–383, 1997.

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

[BLV06] Boaz Barak, Yehuda Lindell, and Salil P. Vadhan. Lower bounds for non-black-box zero knowledge. *J. Comput. Syst. Sci.*, 72(2):321–391, 2006.

[BP04a] Mihir Bellare and Adriana Palacio. The knowledge-of-exponent assumptions and 3-round zero-knowledge protocols. In *Proceedings of the 24th Annual International Cryptology Conference*, pages 273–289, 2004.

[BP04b] Mihir Bellare and Adriana Palacio. Towards plaintext-aware public-key encryption without random oracles. In *ASIACRYPT*, pages 48–62, 2004.

[BP13a] Nir Bitansky and Omer Paneth. On the impossibility of approximate obfuscation and applications to resettable cryptography. In *STOC*, pages 241–250, 2013.

[BP13b] Elette Boyle and Rafael Pass. Limits of extractability assumptions with distributional auxiliary input. *IACR Cryptology ePrint Archive*, 2013:703, 2013.

[BR13] Zvika Brakerski and Guy Rothblum. Virtual black-box obfuscation for all circuits via generic graded encoding. *IACR Cryptology ePrint Archive*, 2013:563, 2013.

[BSW12] Dan Boneh, Gil Segev, and Brent Waters. Targeted malleability: homomorphic encryption for restricted computations. In *ITCS*, pages 350–366, 2012.

[BV11] Zvika Brakerski and Vinod Vaikuntanathan. Efficient fully homomorphic encryption from (standard) lwe. In *FOCS*, pages 97–106, 2011.

[BW13] Dan Boneh and Brent Waters. Constrained pseudorandom functions and their applications. *IACR Cryptology ePrint Archive*, 2013:352, 2013.

[BZ13] Dan Boneh and Mark Zhandry. Multiparty key exchange, efficient traitor tracing, and more from indistinguishability obfuscation. *IACR Cryptology ePrint Archive*, 2013:642, 2013.

[CD08] Ran Canetti and Ronny Ramzi Dakdouk. Extractable perfectly one-way functions. In *Proceedings of the 35th International Colloquium on Automata, Languages and Programming*, pages 449–460, 2008.

[CD09] Ran Canetti and Ronny Ramzi Dakdouk. Towards a theory of extractable functions. In *TCC*, pages 595–613, 2009.

[CLT13] Jean-Sébastien Coron, Tancrède Lepoint, and Mehdi Tibouchi. Practical multilinear maps over the integers. In *CRYPTO (1)*, pages 476–493, 2013.

[COSV12] Chongwon Cho, Rafail Ostrovsky, Alessandra Scafuro, and Ivan Visconti. Simultaneously resettable arguments of knowledge. In *TCC*, pages 530–547, 2012.

[CP13] Kai-Min Chung and Huijia Linand Rafael Pass. Constant-round concurrent zero knowledge from p-certificates. In *FOCS*, 2013.

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

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

[DFH12] Ivan Damgård, Sebastian Faust, and Carmit Hazay. Secure two-party computation with low communication. In *TCC*, pages 54–74, 2012.

[DN07] Cynthia Dwork and Moni Naor. Zaps and their applications. *SIAM J. Comput.*, 36(6):1513–1543, 2007.

[DOP05] Yevgeniy Dodis, Roberto Oliveira, and Krzysztof Pietrzak. On the generic insecurity of the full domain hash. pages 449–466, 2005.

[FLS99] Uriel Feige, Dror Lapidot, and Adi Shamir. Multiple noninteractive zero knowledge proofs under general assumptions. *SIAM J. Comput.*, 29(1):1–28, 1999.

[FS90] Uriel Feige and Adi Shamir. Witness indistinguishable and witness hiding protocols. In *STOC*, pages 416–426, 1990.

[GGH13a] Sanjam Garg, Craig Gentry, and Shai Halevi. Candidate multilinear maps from ideal lattices. In *EUROCRYPT*, pages 1–17, 2013.

[GGH+13b] Sanjam Garg, Craig Gentry, Shai Halevi, Mariana Raykova, Amit Sahai, and Brent Waters. Candidate indistinguishability obfuscation and functional encryption for all circuits. In *FOCS*, 2013.

[GGHR13] Sanjam Garg, Craig Gentry, Shai Halevi, and Mariana Raykova. Two-round secure mpc from indistinguishability obfuscation. *IACR Cryptology ePrint Archive*, 2013:601, 2013.

[GGM86] Oded Goldreich, Shafi Goldwasser, and Silvio Micali. How to construct random functions. *J. ACM*, 33(4):792–807, 1986.

[GGPR13] Rosario Gennaro, Craig Gentry, Bryan Parno, and Mariana Raykova. Quadratic span programs and succinct nizks without pcps. In *EUROCRYPT*, pages 626–645, 2013.

[GHV10] Craig Gentry, Shai Halevi, and Vinod Vaikuntanathan. *i*-hop homomorphic encryption and rerandomizable yao circuits. In *CRYPTO*, pages 155–172, 2010.

[GK96] Oded Goldreich and Hugo Krawczyk. On the composition of zero-knowledge proof systems. *SIAM J. Comput.*, 25(1):169–192, 1996.

[GK05] Shafi Goldwasser and Yael Tauman Kalai. On the impossibility of obfuscation with auxiliary input. In *FOCS*, pages 553–562, 2005.

[GLR11] Shafi Goldwasser, Huijia Lin, and Aviad Rubinstein. Delegation of computation without rejection problem from designated verifier CS-proofs. Cryptology ePrint Archive, Report 2011/456, 2011.

[GMR89] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proof systems. *SIAM J. Comput.*, 18(1):186–208, 1989.

[GMW87] Oded Goldreich, Silvio Micali, and Avi Wigderson. How to play any mental game. In *STOC '87: Proceedings of the nineteenth annual ACM symposium on Theory of computing*, pages 218–229, 1987.

[GO94] Oded Goldreich and Yair Oren. Definitions and properties of zero-knowledge proof systems. *Journal of Cryptology*, 7(1):1–32, December 1994.

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

[Gro10] Jens 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*, pages 321–340, 2010.

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

[HH09] Iftach Haitner and Thomas Holenstein. On the (im)possibility of key dependent encryption. pages 202–219, 2009.

[HK12] Shai Halevi and Yael Tauman Kalai. Smooth projective hashing and two-message oblivious transfer. *J. Cryptology*, 25(1):158–193, 2012.

[HSW13] Susan Hohenberger, Amit Sahai, and Brent Waters. Replacing a random oracle: Full domain hash from indistinguishability obfuscation. *IACR Cryptology ePrint Archive*, 2013:509, 2013.

[HT98] Satoshi Hada and Toshiaki Tanaka. On the existence of 3-round zero-knowledge protocols. In *Proceedings of the 18th Annual International Cryptology Conference*, pages 408–423, 1998.

[KPTZ13] Aggelos Kiayias, Stavros Papadopoulos, Nikos Triandopoulos, and Thomas Zacharias. Delegatable pseudorandom functions and applications. *IACR Cryptology ePrint Archive*, 2013:379, 2013.

[KRR14] Yael Tauman Kalai, Ran Raz, and Ron D. Rothblum. How to delegate computations: The power of no-signaling proofs. In *STOC*, 2014.

[KRW13] Venkata Koppula, Kim Ramchen, and Brent Waters. Separations in circular security for arbitrary length key cycles. *IACR Cryptology ePrint Archive*, 2013:683, 2013.

[Lip12] Helger Lipmaa. Progression-free sets and sublinear pairing-based non-interactive zero-knowledge arguments. In *TCC*, pages 169–189, 2012.

[Lip13] Helger Lipmaa. Succinct non-interactive zero knowledge arguments from span programs and linear error-correcting codes. *IACR Cryptology ePrint Archive*, 2013:121, 2013.

[LS90] Dror Lapidot and Adi Shamir. Publicly verifiable non-interactive zero-knowledge proofs. In *CRYPTO*, pages 353–365, 1990.

[Mic00] Silvio Micali. Computationally sound proofs. *SIAM Journal on Computing*, 30(4):1253–1298, 2000. Preliminary version appeared in FOCS '94.

[Mie08] Thilo Mie. Polylogarithmic two-round argument systems. *Journal of Mathematical Cryptology*, 2(4):343–363, 2008.

[MR13] Tal Moran and Alon Rosen. There is no indistinguishability obfuscation in pessiland. *IACR Cryptology ePrint Archive*, 2013:643, 2013.

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

[NP01] Moni Naor and Benny Pinkas. Efficient oblivious transfer protocols. In *SODA*, pages 448–457, 2001.

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

[PW09] Rafael Pass and Hoeteck Wee. Black-box constructions of two-party protocols from one-way functions. In *TCC*, pages 403–418, 2009.

[SW14] Amit Sahai and Brent Waters. How to use indistinguishability obfuscation: Deniable encryption, and more. In *STOC*, 2014.

[Yao86] Andrew Chi-Chih Yao. How to generate and exchange secrets (extended abstract). In *FOCS*, pages 162–167, 1986.

A Black-Box Lower Bounds

In our construction of EOWFs (or GEOWFs) against bounded-auxiliary-input adversaries, the extractor is non-black-box, i.e., 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 (a similar implication can be shown for the case of generalized EOWFs). 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 large enough security parameter $n \in \mathbb{N}$, and any advice $z \in \{0,1\}^{b(n)}$:

$$\Pr_{e \leftarrow \mathcal{K}_{\mathcal{F}}(1^n)} \left[\begin{array}{c} y \leftarrow \mathcal{M}(e;z) \\ \exists x : f_e(x) = y \end{array} \wedge \begin{array}{c} x' \leftarrow \mathcal{E}^{\mathcal{M}(\cdot;z)}(e) \\ f_e(x') \neq y \end{array} \right] \leq \mathrm{negl}(n) \ .^4$$

This essentially follows the same idea behind the impossibility presented in Section 3, 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 pseudo-random function 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 $y = f_e(x)$. Using the guarantee of the pseudo-random function, it is not hard to see that any black-box extractor $\mathcal{E}$ can be used to break the one-wayness property of $\mathcal{F}$ (using a much simplified version of the proof in Section 3).

Note that the above does not hold when $b(n) = O(\log(n))$, since then the advice cannot contain a seed for a secure pseudo-random function. 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 is guaranteed that one of these executions outputs a valid preimage.

3-round ZK with black-box simulation. Goldreich and Krawczyk [GK96] show that a 3-message protocol for a language $\mathcal{L} \notin$ BPP that is zero-knowledge against non-uniform 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 non-uniform 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 polytime simulator, we must require simulation for verifiers with advice of arbitrary polynomial length.

However, assuming one-way functions exist, we can replace the p-wise independent hash function in the construction of $\mathcal{V}$ by a pseudo-random function with seed length that is independent of p. Then, using the same argument as [GK96], we 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 EOWF, there is no impossibility for 3-message ZK against b-bounded auxiliary-input verifiers 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-round ZK. Goldreich and Oren [GO94] show that 2-message protocols for any language $\mathcal{L} \notin$ BPP that are zero-knowledge against non-uniform 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 can in fact be extended 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] sice it is only ZK against b-bounded auxiliary-inputadversaries where b is smaller then the length of the first protocol message.