How To Construct Extractable One-Way Functions Against Uniform Adversaries

Nir Bitansky[*], Ran Canetti[†], and Omer Paneth[‡]

Tel Aviv University and Boston University

August 1, 2013

Abstract

A function f is extractable if it is possible to algorithmically "extract," from any 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 give the first construction of extractable one-way functions assuming only standard hardness assumptions (e.g., subexponential security of Decision Diffie-Hellman or Quadratic Residousity). Our functions are extractable against adversaries with bounded polynomial advice and unbounded polynomial running time. 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.

The construction uses ideas from [Barak, FOCS01] and [Barak, Lindell, and Vadhan, FOCS03], and rely on the recent breakthrough construction of privately verifiable P-delegation schemes [Kalai, Raz, and Rothblum]. The extraction procedure uses the program evaluating f in a non-black-box way, which we show to be necessary.

1 Introduction

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

One basic flavor of such extraction takes the form of security reductions; namely, efficiently transforming an adversarial program that breaks the security of the analyzed scheme into a program that performs some computational task that is believed to be hard. Examples include transforming a distinguisher for a pseudorandom generator into an inverter of the underlying one-way function [BM84, GL89, HILL99], or transforming a cheating prover in a proof system into a collision-finder in the underlying function [Kil92].

[*]Supported by an IBM Ph.D. Fellowship, the Check Point Institute for Information Security, and an ISF grant 20006317. Part of this research was conducted while visiting Boston University.

[†]Supported by the Check Point Institute for Information Security, an ISF grant 20006317, and an NSF grant 1218461.

[‡]Supported by the Simons award for graduate students in theoretical computer science and NSF award 1218461. Part of this research was conducted while at Microsoft Research New England.

Adversarial knowledge also plays a central role in formulating definitions of security: Here we often require that adversarial participants in a computation "know" their inputs to the computation. This is used as a means for guaranteeing global meaningfulness and secure composition of protocols, for instance in the context of proofs of knowledge and ideal-process based security [FS89, Can00, Can01].

The ability to extract values from the adversary is also useful when *simulating* the adversary's view in a given protocol, to establish privacy of secret inputs, as in the case of zero-knowledge or multi-party computation [GMR89, GMW87]. A quintessential example here is the Feige-Lapidot-Shamir paradigm [FLS99], discussed in more detail below and used extensively in the protocol design literature.

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

Starting with the work of Barak et al. [Bar01], a handful of extraction techniques that go beyond the limitations of black-box extraction have been developed. These techniques use in an essential way having access to the actual adversarial program. However, these techniques too require at least several rounds of protocol interaction. Thus, like black-box rewinding techniques, do not work in the above contexts.

Extractable functions. Originating in Damgård's work [Dam92], and abstracted by Canetti and Dakdouk [CD08, CD09], the notion of *extractable functions* provides an alternative extraction method that does not rely on multiple rounds of interaction with the adversary. These are function families $\{f_k\}$ where, in addition to standard hardness properties, such as one-wayness or collision-resistance, any program M that given k outputs v in the image of f_k has an "extractor" Ext that given k and the code of M, outputs a preimage of v. As an expression to their power, extractable one-way functions are known to suffice for constructing 3-message zero-knowledge protocols [HT98, BP04, CD09]. Extractable collision-resistant hash functions are known to suffice for constructing succinct non-interactive arguments (SNARGs) [BCCT12]. Extractable functions also given rise to relatively efficient CCA constructions [Dam92, BP04].

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 M only as a black box. Coming up with the suitable non-black-box techniques has been the main obstacle in constructing extractable function, 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]; namely, unlike standard assumptions where it possible to algorithmically study the best possible "breakers", here we do not even have an algorithmic way to test whether a given adversary M 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 useful extractable functions from standard hardness assumptions?

1.1 Results

We provide a positive answer to this question. Specifically, we construct one-way functions that are extractable against adversaries with non-uniform advice of bounded polynomial length, but unbounded polynomial running time (from hereon, BAPT adversaries). Note that this class includes, in particular, all uniform PT adversaries. We also show how to use our constructions to obtain 3-message and 2-message zero-knowledge, against BAPT adversarial verifiers.

A bit more precisely, we construct a variant of EOWF, which we call *generalized extractable one-way functions* (GEOWF). A GEOWF is associated with an equivalence relation on its range. The one-wayness requirement is strengthened: not only is it hard to find an exact preimage of v, but it is also hard to find a preimage of any equivalent v'. The extractability requirement is weakened commensurately: the extractor does not have to output a preimage of v, but only a preimage of some equivalent v'. Such a generalization was previously considered in [BCCT12] for a similar purpose, in the context of extractable collision-resistant hash functions (referred to as proximity ECRHs).

Our extraction technique is inspired by the uniform public-coin ZK protocol of Barak [Bar01].[1] The main technical tool used in our constructions are non-interactive computationally sound proofs for deterministic polytime statements, from hereon referred to as P-delegation schemes. In a recent breakthrough, Kalai, Raz, and Rothblum [KRR] construct a P-delegation scheme based on the assumption of a subexponentially secure private information retrieval scheme.

We show:

Theorem 1.1 (informal)**.**

1. *Assuming P-delegation, and 1-Hop homomorphic encryption with function privacy [GHV10], there exist GEOWFs against BAPT adversaries.*

2. *Assuming GEOWFs as above and ZAPs [DN07], there exist a 3-message ZK argument of knowledge against BAPT verifiers.*

3. *Assuming the GEOWFs are one-way against subexponential adversaries, there exists a 2-message ZK argument against BAPT verifiers.*

We note that all the assumptions above can be reduced, for example, to subexponential Decision Diffie-Hellman (or Quadratic Residuosity) and trapdoor permutations [AIR01, NP01, DN07, OI07, GHV10, HK12, KRR].

Limitations and previous work. 3-message zero-knowledge protocols with black-box simulation exist only for trivial languages [GK96]. 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. All previous 3-message zero-knowledge protocols were based on knowledge of exponent (or more general extractability) assumptions, where the simulator uses a non-black extractor that is only assumed to exist, but not explicitly constructed [HT98, BP04, CD08].

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 et al. [BLV06] construct a 2-message argument that is zero-knowledge as long as the verifier's advice is shorter than the verifier message by super-logarithmic additive factor. However, security of the Barak et al. protocol is only shown assuming existence of P-delegation [2] schemes that

[1] The term "uniform ZK" is used in Barak [Bar01]. Barak, Lindell, Vadhan [BLV06] use the term "plain ZK" instead, to stress the fact that only the verifier is uniform, while all other components (including prover and distinguisher) are non-uniform.

[2] In fact, [BLV06] assume non-interactive CS proofs for all of NP; however, their argument can be augmented and based on CS proofs for P

are *publicly verifiable*. While this assumption is falsifiable [CLP13], its only candidate constructions are either Micali's CS proof construction in the random oracle model, or the publicly verifiable SNARGs of [BCCT13] based on knowledge of exponent type assumptions. We obtain the same result, under the assumptions stated above.

1.2 Techniques

Constructing extractable functions. We now sketch how we construct extractable one-way functions against BAPT adversaries. To convey the basic idea behind our construction, consider the following first attempt. Let $\mathsf{PRG} : \{0,1\}^n \to \{0,1\}^{3n}$ be a pseudorandom generator. Define $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^{3n}$ as so:

$$f(i;s) = \begin{cases} \mathsf{PRG}(s) & \text{if } i \neq 0^n \\ s(1^n) & \text{if } i = 0^n \end{cases}.$$

That is, if $i \neq 0^n$, apply the PRG to the seed s. Otherwise, interpret s as a machine, and output its first $3n$ output bits. The one-wayness of f follows from the pseudo-randomness of its output and the fact that a truly random output would have high Kolmogorov complexity. f is also extractable with respect to programs M whose description size is bounded by n: If M outputs some $v \in \{0,1\}^{3n}$, the extractor Ext just outputs $(0^n, M)$.

The main problem is that the time required to compute f is not bounded by any particular polynomial. One can try to fix this by padding the input with 1^t where t is the running time of $s(1^n)$. However, now the length of the extracted primage depends on the running time of M and is not bounded by any particular polynomial in the length of the image. Such 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 [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 v of $s(1^n)$, f will (quickly) verify a proof for the fact that $s(1^n)$ outputs v. That is, we define $f'(0^n, s, v, \pi)$ that outputs v only if π is a convincing proof that $s(1^n) = v$. Intuitively, the soundness of the proof guarantees that function is still one-way. Extraction from a BAPT adversary M is done by simply computing a proof for its computation.

However, which proof system should we use? Barak [Bar01] uses interactive *universal arguments* [Kil92, BG08], whereas we need a non-interactive version of universal arguments that we call NIUA. More precisely, in a NIUA, 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 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).

EOWFs can be constructed from NIUA that are *publicly verifiable*, that is, where the verification state τ can be published without compromising soundness. We define a function family $\{f'_k\}$ as above, where the the key k contains the NIUA offline message σ and the verification state τ. The function f'_k will verify an NIUA proof π for the fact that the program s outputs v within, say, $n^{\log n}$ steps.

Instantiating NIUAs. Alas, we do not know how to construct publicly verifiable NIUAs from standard assumptions. Still, in a recent breakthrough result, Kalai, Raz and Rothblum [KRR13, KRR] construct P-delegation scheme based on any private information retrieval scheme with sub-exponential security. Such P-delegation scheme directly yield NIUA that are *privately-verifiable*. The problem is that now,

including the NIUA's private verification state τ as part of the key k will break the soundness of the NIUA and thus also the one-wayness of f'_k.

Our solution is to modify the function as follows: The key k will contain the offline message σ together with a fully homomorphic encryption c_τ of the corresponding verification state τ. (In fact, a 1-HOP, non-compact homomorphic encryption suffices.) Intuitively, the function is defined as before except that verification of the proof π is done homomorphically, using the encrypted verification state τ. That is, we define $f''_{\sigma,\mathsf{c}_\tau}(0^n, s, v, \pi) = (v, \hat{\mathsf{c}})$, where $\hat{\mathsf{c}}$ is the result of homomorphically evaluating the privately-verifiable NIUA verifier. In the case that the first field i is not 0^n, instead of just outputting $\mathsf{PRG}(s)$, the output includes also an encryption c_1 of the value 1.

Generalized EOWF. Note that the function f'' is no longer extractable in the standard sense. Given a BAPT adversary M that outputs an image $(v, \hat{\mathsf{c}})$, the extractor can compute a proof π for the computation of M and output the preimage $(0^n, M, v, \pi)$ as before. However, f'' on this preimage will output $(v, \hat{\mathsf{c}}')$ where $\hat{\mathsf{c}}'$ is an encryption of 1 that is most probably different than $\hat{\mathsf{c}}$. Nonetheless, f'' satisfies the notion of generalized extractable one-way functions (GEOWF). We define an equivalence relation $\sim$ on the range of f_k as follows: $(v, \hat{\mathsf{c}}) \sim (v', \hat{\mathsf{c}}')$ iff $v = v'$ and both $\hat{\mathsf{c}}$ and $\hat{\mathsf{c}}'$ are encryptions of 1. As required, the extractor described above outputs a preimge of $(v, \hat{\mathsf{c}}')$ such that $(v, \hat{\mathsf{c}}) \sim (v, \hat{\mathsf{c}}')$. The one-wayness of f'' with respect to $\sim$ follows from the one-wayness of f'_k and the semantic security of the encryption: given an image $(v, \hat{\mathsf{c}})$, finding a primage u of $(v, \hat{\mathsf{c}}')$ such that $(v, \hat{\mathsf{c}}) \sim (v, \hat{\mathsf{c}}')$ is hard, since if $\hat{\mathsf{c}}'$ is an encryption of 1, u must also be a valid preimage of v under f', whereas by sematic security the encrypted verification state c_τ should not help in inverting f'. Note that the relation $\sim$ can only be tested given the secret key for the encryption. We refer to such a relation as privately testable. (The above is somewhat of an oversimplification of the actual definition of f''_k. See the body for details.)

Next, we show how to construct 2-message and 3-message zero-knowledge protocols from GEOWF with a privately testable relation.

From EOWF to 3-message zero-knowledge. We start by describing a 3-message zero-knowledge protocol from strict EOWFs (namely, when the equivalence relation $\sim$ is equality). The protocol follows the Feige-Lapidot-Shamir *trapdoor paradigm* [FLS99]. The basic idea is to have the verifier sample a function f_k from the EOWF family, and send an image $v = f_k(u)$ of a random element u, 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 u' of v. Intuitively, soundness (and actually proof of knowledge) follow from the one-wayness of f_k and the proof of knowledge property of the WI system. Zero-knowledge follows from the extractability of f_k. Indeed, the simulator, given the code of the verifier, can run the extractor of the EOWF, obtain u, and use it to simulate the WI proof.

Following through on this intuition encounters several difficulties. First, since a WI proof of knowledge requires 3 messages, the first WI prover message must be sent in the first message of the protocol. However, the WI statement is only determined when the verifier sends v 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 BAPT adversaries, the function's output is longer than the adversary's advice. Since a cheating verifier may compute v using the first WI message as an advice, we must use a WI system where the length of the first message is independent of the length of the proven statement. We design a WI proof of knowledge with the required properties based on ZAPs [DN07] and extractable commitments [PW09].

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. 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 "$f_k(u) = v$ *and the commitment is to* u". We require that the commitment is invertible in some superpolynomial time T, whereas the one-wayness of f_k still holds against adversaries that run in time $\mathrm{poly}(T)$. Now, an inverter of f_k can run the cheating prover with a verifier message that contains its input image v, and brute-force break the commitment to obtain a preimage of v.

Replacing EOWF with GEOWF. The zero-knowledge protocols from GEOWFs are similar, with the exception that now, rather than proving knowledge of a preimage of v, the prover will send an image v' such that $v' \sim v$ and prove knowledge of a preimage of v'. To keep the protocol ZK, the honest prover must be able to sample an image v' that is distributed like the image that is sent by the simulator even without knowing a corresponding preimage. In our GEOWF construction, such an equivalent image can be efficiently sampled.

1.3 Why Extractable Functions?

As pointed out above, the extractable functions constructed here mimic Barak's zero-knowledge protocol [Bar01]. The similarity becomes even stronger when considering the two-message zero-knowledge protocol of Barak et. al [BLV06]: Our two message protocol can be directly obtained from that of [BLV06] by replacing the CS proofs with P-delegation, and accounting for private verifiability as sketched above. This can be done without mention of extractable functions. Still, we believe that the abstraction of extractable functions is helpful in this context. In particular, it helps separating the protocol structure from the underlying mechanism of extracting a secret value from a given adversarial program.

Furthermore, we hope that this abstraction will prove useful for additional applications beyond two and three-message zero-knowledge. Applications like succinct non-interactive arguments (SNARGs) and efficient CCA encryption seem to require extractable functions with stronger properties such as injectiveness or collision-resistance [Dam92, BCCT12]. At this point, candidates for extractable functions with such properties are known based on non-standard assumptions regarding different number theoretic and algebraic structures, such as the knowledge-of-exponent assumption. In contrast, our construction is unstructured and does not satisfy the above properties. Indeed, in our function it is easy to find collisions: Consider a machine M that just evaluates the function on any arbitrary input u. By simply applying the extractor on M, we can obtain a different preimage u' mapping to an equivalent image.

We hope that the proposed construction will provide a stepping stone to improved constructions of stronger extractable functions based on standard and better understood hardness assumptions. Two natural targets here are extractable collision-resistant hash functions and extractable non-interactive commitments.

Organization. Section 2 defines the notion of NIUA and other main tools used in our constructions. Section 3 presents our main construction of GEOWF against BAPT adversaries. Section 4 constructs 2-message and 3-message ZK protocols against BAPT verifiers.

2 Tools

2.1 Non-Interactive Universal Arguments for Deterministic Computations & Delegation

We define *non-interactive universal arguments for deterministic computations* (NIUAs), which can be seen as a special case of Barak's and Goldreich's [BG08] UAs. We then explain the relation to the problem of delegation and the corresponding existence result of Kalai, Raz, and Rothblum [KRR].

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

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 $\mathcal{P}(M, x; \sigma)$ produces a certificate π for the fact that $(M, x) \in \mathcal{L}_{\mathcal{U}}(T(n))$. The verifier $\mathcal{V}(M, x; \pi, \tau)$ verifies the validity of π.

Definition 2.1 (NIUA)**.** *A triple of algorithms $(\mathcal{G}, \mathcal{P}, \mathcal{V})$ is a non-interactive universal argument system for for Dtime(T) if it satisfies:*

- *Perfect Completeness:*
  For any $n \in \mathbb{N}$ and $(M, x) \in \mathcal{L}_{\mathcal{U}}(T(n))$:
  $$\Pr_{(\sigma,\tau)\leftarrow\mathcal{G}(1^n)}[\mathcal{V}(M, x; \pi, \tau) = 1 \mid \pi \leftarrow \mathcal{P}(M, x; \sigma)] = 1 \ .$$
- *Adaptive soundness:*
  For any polysize prover $\mathcal{P}^*$ and large enough $n \in \mathbb{N}$:
  $$\Pr_{(\sigma,\tau)\leftarrow\mathcal{G}(1^n)}\left[\mathcal{V}(M, x; \pi, \tau) = 1 \ \middle| \ \begin{array}{c} (M, x, \pi) \leftarrow \mathcal{P}^*(\sigma) \\ (M, x) \in \{0,1\}^n \setminus \mathcal{L}_{\mathcal{U}}(T(n)) \end{array} \right] \leq \mathrm{negl}(n) \ .$$
- *Fast verification and relative prover efficiency:*
  There exists a polynomial p such that for every $n \in \mathbb{N}$, $t \leq T(n)$, and $(M, x) \in \mathcal{L}_{\mathcal{U}}(t)$:
  - *the generator $\mathcal{G}$ runs in time $p(n)$;*
  - *the verifier $\mathcal{V}$ runs in time $p(n + |M| + |x|)$;*
  - *the prover $\mathcal{P}$ runs in time $p(n + |M| + |x| + t)$.*

The system is said to be publicly verifiable *if it is sound when τ is public. In this case, we will assume WLOG that $\sigma = \tau$.*

Existence and connection to delegation of computation. There are two differences between the notion of delegation for deterministic computations (See, e.g., [KRR13]) and the NIUA notion defined above. The first is that a delegation system is associated with a given language $\mathcal{L}(M)$ for a fixed deterministic machine M, and the corresponding efficiency parameters depend on the worst-case running time T_M of M. In particular, the generator $\mathcal{G}$ depends on T_M as an extra parameter, and the prover's efficiency is polynomial in the worst-case running time T_M. The second difference is that only non-adaptive soundness is required; in particular, the generator's message σ may depend on the input x.

Kalai, Raz, and Rothblum [KRR] 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 that are subexponentially secure, which can in turn be constructed based the subexponential Learning with Errors assumption [BV11].[3]

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 M with running time $t_M << T$. 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

[3]Private information retrieval based on Quadratic Residousity, Decision Diffie-Helmman, or more generally additively homomorphic encryption can also be used. However, this induces worst communication complexity [OI07], and leads to NIUAs for smaller, but still super-polynomial, upper bound T; this is still sufficient for our purpose.

their result. Specifically, we could run the generator in their solution to generate reference string and verification state (σ, τ) for computations of size t for all $t \in \{1, 2, 2^2, \ldots, 2^{\log T}\}$, and have the prover and verifier use the right (σ, τ) according to the concrete running time $t_M < T$, guaranteeing that the prover's running time is at most $\mathrm{poly}(2t_M)$ as required.

Also, in their scheme, the generator works independently of the input x, but soundness is shown only when σ is generated independently of x. To guarantee soundness for adaptively chosen inputs $x \in \{0,1\}^n$, we may repeat the above argument $2n$ times. Since parallel repetition exponentially reduces the soundness error in two-message arguments, we can then take a union bound over all 2^n adaptive choices of x and get the required soundness. The $2n$-factor hit in succinctness and verification time are still tolerable for our purposes (and still satisfy the above definition).

2.2 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 2.2. *A scheme* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Eval}, \mathsf{Dec})$ *is a semantically-secure, circuit-private, 1-Hop homomorphic encryption scheme if it satisfies the following properties:*

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

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

- *Semantic security:*
  For any polysize $\mathcal{A}$*, large enough* $n \in \mathbb{N}$*, and any pair of inputs* $u_0, u_1 \in \{0,1\}^n$

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

- *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:*

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

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

3 Extractable One-Way Functions against Adversaries with Bounded Advice

In this section, we define and construct extractable one-way functions against adversaries with bounded advice.

3.1 Definitions

For brevity, we refer to extractable one-way functions against polytime adversaries with m-bounded advice as EOWFs against m-BAPT adversaries. Such functions are one-way in the usual sense, and in addition it is possible to efficiently extract a pre-image from the code of any adversary that outputs a valid image, provided that the adversary only has bounded non-uniform advice, but arbitrary polynomial running time; this, in particular, includes the class of *uniform polytime adversaries*. Concretely, we shall focus on PPT adversaries with non-uniform advice $z \in \{0,1\}^{m(n)}$. We treat any randomness that the machines may use as part of their advice z; in particular, our adversaries are only allowed bounded randomness, which in most application is not a restriction, as they can use a PRG to stretch it (see Remark 3.2).

Definition 3.1 (EOWFs against m-BAPT adversaries)**.** *Let ℓ, ℓ' be polynomially bounded length functions. An efficiently computable family of functions*

$$\mathcal{F} = \left\{ f_k : \{0,1\}^{\ell(n)} \to \{0,1\}^{\ell'(n)} \;\middle|\; k \in \{0,1\}^{\mathrm{poly}(n)}, n \in \mathbb{N} \right\} \ ,$$

associated with an efficient (probabilistic) key sampler $\mathcal{K}$, is an extractable one-way function against adversaries with m-bounded advice if it satisfies:

- *One-wayness: For any polysize $\mathcal{A}$, and large enough security parameter $n \in \mathbb{N}$:*

$$\Pr_{\substack{k \leftarrow \mathcal{K}(1^n) \\ u \leftarrow \{0,1\}^{\ell(n)}}} \left[\begin{array}{c} u' \leftarrow \mathcal{A}(k, f_k(u)) \\ f_k(u') = f_k(u) \end{array} \right] \leq \mathrm{negl}(n) \ .$$

- *Extractability: For any PPT adversary M, there exists a PPT extractor Ext such that, for any large enough security parameter $n \in \mathbb{N}$, and advice $z \in \{0,1\}^{m(n)}$:*

$$\Pr_{k \leftarrow \mathcal{K}(1^n)} \left[\begin{array}{ccc} v \leftarrow M(k;z) & \wedge & u' \leftarrow \mathsf{Ext}(k;z) \\ \exists u : f_k(u) = v & & f_k(u') \neq v \end{array} \right] \leq \mathrm{negl}(n) \ .$$

Generalized EOWFs. We next define *generalized EOWFs* (GEOWFs), analogous to the *proximity extractable collision-resistance* in [BCCT12]. Here the extractable function family $\{f_k\}$ is associated with an equivalence relation $\sim$ on the range of f_k. The one-wayness requirement is then strengthened: not only is it hard to find an exact preimage of v, but it is also hard to find a preimage of any equivalent $v' \sim v$ (we shall often refer to such a preimage as a "relative preimage"). The extractability requirement is weakened accordingly: the extractor does not have to output an exact preimage of v, but only a preimage of some equivalent $v' \sim v$. Intuitively, one can think of a generalized EOWF where each f_k maps $\{0,1\}^{\ell}$ to $\{0,1\}^{\ell'}$ as a standard EOWF that maps $\{0,1\}^{\ell}$ to the quotient space of co-sets in $\{0,1\}^{\ell'}$ modulo $\sim$.

We may further allow that the relation $\sim$ depends on the key k, where the relation $\sim$ can either be publicly-testable given k, or require the private coins used to sample k for efficient testing. In particular, any standard EOWF is a GEOWF with the publicly-testable equality relation.

Definition 3.2 (GEOWFs against m-BAPT adversaries)**.** *An efficiently computable family of functions*

$$\mathcal{F} = \left\{ f_k : \{0,1\}^{\ell(n)} \to \{0,1\}^{\ell'(n)} \;\middle|\; k \in \{0,1\}^{\mathrm{poly}(n)}, n \in \mathbb{N} \right\} \ ,$$

associated with an efficient (probabilistic) key sampler $\mathcal{K}$, is an extractable one-way function, with proximity relation $\sim$ on $\{0,1\}^{\ell'(n)} \times \{0,1\}^{\ell'(n)}$, against adversaries with m-bounded advice, if it satisfies:

- *Strong one-wayness: for any polysize $\mathcal{A}$, and large enough security parameter $n \in \mathbb{N}$:*

$$\Pr_{\substack{k \leftarrow \mathcal{K}(1^n) \\ u \leftarrow \{0,1\}^{\ell(n)}}} \left[\begin{array}{c} u' \leftarrow \mathcal{A}(k, f_k(u)) \\ f_k(u) \sim f_k(u') \end{array} \right] \leq \mathrm{negl}(n) \ .$$

- *Weak extractability: for any PPT adversary M, there exists a PPT extractor* Ext *such that, for any large enough security parameter $n \in \mathbb{N}$, and advice $z \in \{0,1\}^{m(n)}$:*

$$\Pr_{k \leftarrow \mathcal{K}(1^n)} \left[\begin{array}{ccc} v \leftarrow M(k;z) & \wedge & u' \leftarrow \mathsf{Ext}(k;z) \\ \exists u : f_k(u) = v & & f_k(u') \not\sim v \end{array} \right] \leq \mathrm{negl}(n) \ .$$

- *Testable relation: There exists a deterministic polytime machine $\mathcal{T}$ such that, given the random coins used by $\mathcal{K}$ to sample k, $\mathcal{T}$ accepts $v, v' \in \{0,1\}^{\ell'(n)}$ if and only if $v \sim_k v'$. The relation is publicly-testable $\mathcal{T}$ only requires k, but not the private coins used to sample it.*

Remark 3.1 (universal extractor). In the above definitions, each PPT M is required to have a designated PPT extractor Ext_M. Our constructions will, in fact, guarantee the existence of one universal extractor Ext that given any (M, z, k) and a bound 1^{t_M} on the running time of $M(k; z)$, can perform extraction. Moreover, the running time of Ext is bounded by some (universal) polynomial $\mathrm{poly}(t_M)$ in the running time of M.

Remark 3.2 (bounded randomness). In our definitions, we assumed that any randomness used by the machines is part of their bounded advice, and in particular, is bounded itself. 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. (Alternatively, we can achieve an alternative extraction definition where the extractor is randomized and is allowed to simulate the adversary's randomness.) This approach is applicable for example for ZK against BAPT verifiers (see Section 4), as well as for any application where testing if the adversary breaks the scheme can be done efficiently.

Additional properties of GEOWFs. We next discuss additional properties of GEOWFs that will prove useful in the design of protocols.

The first property is called *everywhere extractability* and requires that extractability does not only hold with overwhelming over keys $k \leftarrow \mathcal{K}(1^n)$, but also holds for any maliciously chosen key.

Definition 3.3. *A GEOWF (or EOWF) is said to satisfy everywhere extractability if weak extractability holds for any $k \in \{0,1\}^{\mathrm{poly}(n)}$, where $\mathrm{poly}(n)$ is the length of keys output by $\mathcal{K}(1^n)$.*

We note that everywhere extractability cannot be achieved if the output of the function ℓ' is shorter than the length of the advice m that the adversary is allowed; otherwise, the adversary may simply get as advice a random image under some key, and the extractor would fail to invert for that key. Indeed, in our constructions that achieve everywhere extractability, the output of the function will be longer than the allowed advice.

Also, we note that one can make an analogous property of *everywhere one-wayness*, which for some applications may be a sufficient alternative to everywhere extractability; however, we will not require (nor achieve) this property in this work.

The second property, called *oblivious image sampling*, strengthens even further the extractability requirement, saying that the distribution of images $v' = f_k(u')$ induced by the extraction procedure Ext is simulatable, given only the image v output by the adversary M (and without the code of M). Accordingly, the extractor Ext will now be randomized.

Definition 3.4. *A baGEOWF with proximity relation $\sim$ is said to satisfy oblivious image sampling if there exists PPT image sampler $\mathcal{I}$ that, given k and $v \in \{0,1\}^{\ell'(n)}$, outputs $\tilde{v} \sim v$ such that for any m-bounded advice adversary M, and corresponding extractor* Ext*:*

$$\left\{ \tilde{v} \;\middle|\; \begin{array}{r} v \leftarrow M(k;z) \\ \tilde{v} \leftarrow \mathcal{I}(k,v) \end{array} \right\}_{\substack{n\in\mathbb{N}, z\in\{0,1\}^{m(n)} \\ k\in\{0,1\}^{\mathrm{poly}(n)}}} \approx_c \left\{ v' \;\middle|\; \begin{array}{r} u' \leftarrow \mathsf{Ext}(k;z) \\ v' = f_k(u') \end{array} \right\}_{\substack{n\in\mathbb{N}, z\in\{0,1\}^{m(n)} \\ k\in\{0,1\}^{\mathrm{poly}(n)}}},$$

where $\mathrm{poly}(n)$ *is the length of keys output by* $\mathcal{K}(1^n)$.

The requirement that $\tilde{v} \sim v$ is stated for the sake of clarity, and actually follows automatically from the fact that everywhere extraction guarantees $v' \sim v$, and that $v' \approx_c \tilde{v}'$. We also note that oblivious image sampling automatically holds for standard EOWFs (rather than GEOWF) that are everywhere extractable. Indeed, the image sampler can just output the same image v it gets as input.

3.2 Constructions

We start by describing a construction of a (standard) EOWF against BAPT adversaries from any publicly verifiable NIUA system and any pseudorandom generator. We then provide an augmented construction of a GEOWF based only on privately verifiable NIUA and any 1-Hop homomorphic encryption scheme.

3.2.1 EOWF from publicly verifiable NIUA.

In what follows, let $(\mathcal{G}, \mathcal{P}, \mathcal{V})$ be a publicly verifiable NIUA system for $\mathrm{Dtime}(T(n))$ for some function $T(n) \in (2^{\omega(\log n)}, 2^{\mathrm{poly}(n)})$, and assume that for security parameter n, $\mathcal{G}(1^n)$ outputs the public reference string (which is also the verification state) $\sigma \in \{0,1\}^n$, and that $\mathcal{P}$ outputs certificates π of size n. Let PRG be a pseudo random generator stretching n bits to $m(n) + 2n$ bits. The construction basically follows the high-level ideas presented in the introduction (with slight syntactic differences).

Construction 3.1. The key generation algorithm $\mathcal{K}$ is just the generator $\mathcal{G}$.

The function is defined as follows:

$$(i; M, y, \pi; s) \xmapsto{f_\sigma} \begin{cases} y, \mathcal{V}(M, \sigma, y; \pi, \sigma) & \text{if } i = 0^n \\ \mathsf{PRG}(s), 1 & \text{if } i \neq 0^n \end{cases},$$

where $\mathcal{V}(M, \sigma, y; \pi, \sigma)$ is the result of verifying the certificate π for the statement "$M(\sigma) = y$ within $T(n)$ steps". Each of the inputs to the function is of length n, except M that is of length m, and y that is of length $m + 2n$, like $\mathsf{PRG}(s)$.

Theorem 3.1. *The function family $\mathcal{F} = \{f_k\}_{n\in\mathbb{N}}$, given by Construction 3.1, is a GEOWF against $(m - \omega(1))$-BAPT adversaries.*

High-level idea behind the proof. To see that the function is one-way, note that a random image v is almost always the output of a PRG (i.e., comes from the $i \neq 0^n$ branch). To invert it, the adversary must either invert the PRG, in which case it can produce a preimage of the $i \neq 0^n$ type, or it could find a short machine M and an accepting proof that it outputs v, in which case it can produce a preimage of the $i = 0^n$ type. The first case does not occur since PRG is one-way. The second case cannot occur because of the pseudorandomness of PRG and the soundness of the NIUA; indeed, had we replace the output v of the PRG with a truly random string, it would almost surely have high Kolomogorov complexity, and a short machine M that outputs v wouldn't exist, meaning that the inverter would have to produce an accepting proof for a false statement, and violate soundness.

As for extraction, given a machine M_z with short advice z that outputs $(v, 1)$, the extractor simply computes a proof π for this computation, and outputs the preimage $(0^n; M_z, v, \pi; 1^n)$. If M outputs

something of the form $(v, 0)$, the extractor produces instead some non accepting statement and proof (M', π'), and outputs $(0^n; M', v, \pi'; 1^n)$.

We omit the full proof, which is subsumed by the proof of the private-verifiability case treated below.

3.2.2 GEOWF from privately verifiable NIUA.

Let $(\mathcal{G}, \mathcal{P}, \mathcal{V})$ be a privately verifiable NIUA system for $\mathrm{Dtime}(T(n))$ for some function $T(n) \in (2^{\omega(\log n)}, 2^{\mathrm{poly}(n)})$, and assume that for security parameter n, $\mathcal{G}(1^n)$ outputs a public reference string σ and private verification state τ, each of size n, and assume that $\mathcal{P}$ outputs certificates π of size n. Let PRG be a pseudo random generator stretching n bits to $m(n) + n^2 + 2n$ bits. Let $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval})$ be a 1-Hop homomorphic encryption scheme (not necessarily compact), and assume that a cipher c encrypting n bits is of size n^2. Let $\mathcal{C}_{\mathcal{V}} = \mathcal{C}_{\mathcal{V}}(M, \sigma, \mathsf{c}, y, \pi, \cdot)$ be a circuit that, given τ as input, verifies that $\mathcal{V}(M, (\sigma, \mathsf{c}), y; \pi, \tau) = 1$; namely, it verifiers the certificate π for the statement "$M(\sigma, \mathsf{c}) = y$ within $T(n)$ steps". Let $\mathcal{C}_{\mathbf{1}}$ be a circuit that always outputs 1 and is of the same size as $\mathcal{C}_{\mathcal{V}}$.

Construction 3.2. The key generator $\mathcal{K}$ samples $(\sigma, \tau) \leftarrow \mathcal{G}(1^n)$, $\mathsf{c}_\tau \leftarrow \mathsf{Enc}(\tau)$, and outputs σ, c_τ.

The function $f_{\sigma, \mathsf{c}_\tau}$ is defined as follows:

$$(i; M, y, \pi; s) \overset{f_{\sigma,\mathsf{c}_\tau}}{\longmapsto} \begin{cases} y, \hat{\mathsf{c}}_{\mathcal{V}}, i & \text{if } i = 0^n \\ \mathsf{PRG}(s), \hat{\mathsf{c}}_1, i & \text{if } i \neq 0^n \end{cases},$$

where $\hat{\mathsf{c}}_{\mathcal{V}} \leftarrow \mathsf{Eval}(\mathcal{C}_{\mathcal{V}}, \mathsf{c}_\tau)$ and $\hat{\mathsf{c}}_1 \leftarrow \mathsf{Eval}(\mathcal{C}_{\mathbf{1}}, \mathsf{c}_\tau)$ (both Eval can be deterministic). Each of the inputs to the function are of length n, except M that is of length m, and y that is of length $m + n^2 + 2n$.

The proximity relation $\sim$ is defined as follows: $(y, \hat{\mathsf{c}}, i) \sim (y', \hat{\mathsf{c}}', i')$ if either

$$\begin{aligned} &y = y' \text{ and } \mathsf{Dec}(\hat{\mathsf{c}}) = \mathsf{Dec}(\hat{\mathsf{c}}') = 1, \\ &\text{or } i = i' = 0 \ . \end{aligned}$$

Theorem 3.2. *The function family $\mathcal{F} = \{f_k\}_{n \in \mathbb{N}}$, given by Construction 3.1, is a GEOWF, against $(m - \omega(1))$-BAPT adversaries, with a privately testable equivalence relation.*

High-level idea behind the proof. The proof follows the same high-level ideas as in the previous construction, except that now there is a private NIUA verification state τ that has to remain secret or soundness, and thus also one-wayness, will be compromised. By including an encryption of τ in the key, we guarantee that soundness and one-wayness are maintained. However, now the NIUA verification has to be done homomorphically under the encryption. The reason that i is added to the output of the function is to guarantee that we can extract in cases that the adversary produces an output of $v, \hat{\mathsf{c}}$, where $\mathsf{Dec}(\hat{\mathsf{c}}) = 0$ (e.g. by choosing a false statement). Unlike in the previous construction, the extractor cannot tell that this is the case simply by looking at $\hat{\mathsf{c}}$, which is why we provide it with i.

We now provide a more detailed proof sketch.

Proof sketch. We first show strong one-wayness, and then show weak extractability. The fact that the relation is testable given the coins used to sample k (specifically the decryption key) follows readily.

Strong one-wayness. Assume that given $f_{\sigma, \mathsf{c}_\tau}(u)$, where $u \leftarrow \{0,1\}^{\ell(n)}$ and $(\sigma, \mathsf{c}_\tau) \leftarrow \mathcal{K}(1^n)$, a polysize inverter $\mathcal{I}$ finds u' such that $f_k(u) \sim f_k(u')$ with noticeable probability ϵ. We describe a polysize adversary $\mathcal{B}$ that breaks the soundness of the NIUA with probability $\epsilon - \mathrm{negl}(n)$. Given σ, $\mathcal{B}$ first samples $\mathsf{c}_0 \leftarrow \mathsf{Enc}(0^n)$, and computes $\hat{\mathsf{c}} \leftarrow \mathsf{Eval}(\mathcal{C}_{\mathbf{1}}, \mathsf{c}_0)$. It then samples $r \leftarrow U_{m+n^2+2n}$, $i \leftarrow U_n$, and runs the inverter $\mathcal{I}((\sigma, \mathsf{c}_0), (r, \hat{\mathsf{c}}, i))$ who outputs $u' = (i'; M', y', \pi'; s')$.

We claim that, with probability $\epsilon - \mathrm{negl}(n)$, π' is a valid certificate for the fact that $M'(\sigma, \mathsf{c}_0) = r$; namely, $\mathcal{V}(M', (\sigma, \mathsf{c}_0), y'; \pi, \tau) = 1$. Indeed, if this is the case, we are done since $|M'| + |\sigma| + |\mathsf{c}_0| \leq$

$m + n + n^2$, and thus $M'(\sigma, \mathsf{c}_0) = r$ with probability at most 2^{-n}. This implies that π' is an accepting proof for a false statement with probability $\epsilon - \mathrm{negl}(n)$. To prove the claim we consider two hybrid breakers.

Breaker $\mathcal{B}_0$ samples $r_0 \leftarrow \mathsf{PRG}(U_n)$, $i \leftarrow U_n$ on its own, gets an external encryption c_τ of the random coins used by the generator $\mathcal{G}$ of the NIUA system, computes $\hat{\mathsf{c}} \leftarrow \mathsf{Eval}(\mathcal{C}_{\mathbf{1}}, \mathsf{c}_\tau)$, and runs $\mathcal{I}((\sigma, \mathsf{c}_\tau), (r_0, \hat{\mathsf{c}}, i))$. We claim that, with probability $\epsilon - \mathrm{negl}(n)$, $\mathcal{B}_0$ obtains u' such that $f_{\sigma,\mathsf{c}_\tau}(u') \sim (r_0, \hat{\mathsf{c}}, i)$. Indeed, $\mathcal{I}$'s input is distributed identically to the original inversion experiment conditioned on $i \neq 0^n$, which occurs with probability $1 - 2^{-n}$.

Now, consider $\mathcal{B}_0$'s output $u' = (i'; M', y', \pi'; s')$. We claim that, except with negligible probability, whenever $(r_0, \hat{\mathsf{c}}, i) \sim f_{\sigma,\mathsf{c}_\tau}(u')$, it must be that $i' = 0^n$ (and $i \neq 0^n$), meaning that $\mathcal{B}_0$ finds (M', y', π'), such that $\mathsf{Dec}(\mathsf{Eval}(\mathcal{C}_\mathcal{V}, \mathsf{c}_\tau)) = \mathsf{Dec}(\hat{\mathsf{c}}) = 1$, and $y' = r_0$, which in turn implies that π' is an accepting proof for the fact that $M'(\sigma, \mathsf{c}_\tau) = y' = r_0$. Indeed, if (relative) inversion occurs with noticeable probability when $i' \neq 0^n$, we can use $\mathcal{B}_0$ to invert PRG, contradicting its one-wayness.

Breaker $\mathcal{B}_1$ operates exactly like $\mathcal{B}_0$, but gets an external encryption c_0 of 0^n, instead of τ. By semantic security, $\mathcal{B}_1$ would also find (M', π') such that π' is a valid for certificate for the statement "$M'(\sigma, \mathsf{c}_0) = r_0$ within $T(n)$ steps" with probability $\epsilon - \mathrm{negl}(n)$.

To conclude the proof, we observe that the only difference between $\mathcal{B}_1$ and $\mathcal{B}$ is that $\mathcal{B}$ samples $r \leftarrow U_{m+n^2+2n}$, rather than $r_0 \leftarrow \mathsf{PRG}(U_n)$, and so the claim follows by pseudo-randomness.

Weak extractability. We now show weak extractability via a universal extractor Ext (see Remark 3.1). For an adversarial code M and advice $z \in \{0,1\}^{m-\omega(1)}$ denote by M_z the machine $M(\cdot; z)$. Given any key $k = (\sigma, \mathsf{c}_\tau)$ and (M, z), where M_z has description size at most $m(n)$ and running time at most $t < T(n)$, and $M_z(\sigma, \mathsf{c}_\tau) = (y, \hat{\mathsf{c}}, i) \in \mathsf{Image}(f_k)$.

Our extractor Ext works according to two cases. If $i = 0^n$, the extractor simply outputs some canonical $u'_* = (i'; M', y', \pi'; s')$, such that $i' = 0^n$. If $i \neq 0^n$, Ext computes a certificate π for the fact that "$M_z(\sigma, \mathsf{c}_\tau) = y$", and then outputs the (relative) pre-image $u' = (0^n, M_z, y, \pi, s)$, where s could be any arbitrary string, e.g., 1^n. By definition $f_k(u') = (y, \mathsf{Eval}(\mathcal{C}_\mathcal{V}, \mathsf{c}_\tau), 0^n)$, so to guarantee that $f_k(u') \sim (y, \hat{\mathsf{c}}, i)$, it is left to see that $\mathsf{Dec}(\mathsf{Eval}(\mathcal{C}_\mathcal{V}, \mathsf{c}_\tau)) = \mathsf{Dec}(\hat{\mathsf{c}})$. Indeed, if $i \neq 0^n$, it is guaranteed that $\mathsf{Dec}(\hat{\mathsf{c}}) = 1$. Furthermore, for any $k \in \mathrm{supp}(\mathcal{K}(1^n))$, by the perfect completeness of the NIUA system and perfect correctness of the 1-Hop homomorphic encryption, it also holds that $\mathsf{Dec}(\mathsf{Eval}(\mathcal{C}_\mathcal{V}, \mathsf{c}_\tau)) = 1$. □

Everywhere extractability and oblivious image sampling. We note that the extractor described above works for $k \in \mathrm{supp}(\mathcal{K}(1^n))$, and not only for an overwhelming fraction of keys in the support. However, the extractor is not guaranteed to work for a maliciously chosen $k \notin \mathrm{supp}(\mathcal{K}(1^n))$. For example, if the NIUA keys σ and τ (determined by c_τ) are maliciously sampled, it may be that the NIUA verification procedure would reject true statements; in particular, it could be that $\mathsf{Dec}(\mathsf{Eval}(\mathcal{C}_\mathcal{V}, \mathsf{c}_\tau)) = 0$, and thus extraction would fail. Looking ahead, the very same problem would prevent us from obtaining oblivious image sampling for maliciously chosen keys.

We now show, however, that we can slightly augment the construction so it would satisfy both everywhere extractability and oblivious image sampling. The high-level idea is to simply embed inside the function a test for the validity of the key k.

Specifically, instead of including in the key only an encryption c_τ of the NIUA's private verification state, we will include an encryption c_r of the randomness r used by the NIUA generator $\mathcal{G}$ to sample (σ, τ). Now, instead of the circuit $\mathcal{C}_\mathcal{V} = \mathcal{C}_\mathcal{V}(M, \sigma, \mathsf{c}, y, \pi, \cdot)$ that, given τ as input, verifies that $\mathcal{V}(M, (\sigma, \mathsf{c}), y; \pi, \tau) = 1$, we will consider a new circuit $\mathcal{C}_{\mathcal{V},\mathcal{G}}(M, \sigma, \mathsf{c}, y, \pi, \cdot)$ that, given r derives τ, and checks the above, but in addition also checks that σ is consistent with $\mathcal{G}(1^n; r)$. Analogously, instead of the circuit $\mathcal{C}_{\mathbf{1}}$ that always outputs 1, we will consider the circuit $\mathcal{C}_\mathcal{G}(\sigma, \cdot)$ that, given r, checks that σ is consistent with $\mathcal{G}(1^n; r)$. A last additional tweak meant to support oblivious image sampling is that now the homomorphic evaluation procedures will be randomized to guarantee circuit privacy.

Overall the augmented function f_{σ,c_r} is defined as follows:

$$\left(i; M, y, \pi; s; r'\right) \overset{f_{\sigma,\mathsf{c}_\tau}}{\longmapsto} \begin{cases} y, \hat{\mathsf{c}}_{\mathcal{V},\mathcal{G}}, i & \text{if } i = 0^n \\ \mathsf{PRG}(s), \hat{\mathsf{c}}_{\mathcal{G}}, i & \text{if } i \neq 0^n \end{cases},$$

where $\hat{\mathsf{c}}_{\mathcal{V}} \leftarrow \mathsf{Eval}(\mathcal{C}_{\mathcal{V}}, \mathsf{c}_r; r')$ and $\hat{\mathsf{c}}_{\mathcal{G}} \leftarrow \mathsf{Eval}(\mathcal{C}_{\mathcal{G}}, \mathsf{c}_\tau; r')$ are now randomized to guarantee circuit privacy. The proximity relation $\sim$ is augmented to account also for the case that $\mathsf{Dec}(\hat{\mathsf{c}}) = 0$ (due to a malicious choice of keys):

$$\begin{aligned} &y = y' \text{ and } \mathsf{Dec}(\hat{\mathsf{c}}) = \mathsf{Dec}(\hat{\mathsf{c}}'), \\ &\text{or } i = i' = 0 \ . \end{aligned}$$

Claim 3.1. *The augmented construction is a GEOWF that satisfies both everywhere extractability and oblivious image sampling.*

Proof sketch. First, since for an honestly sampled k the new function is equivalent to the previous, strong one-wayness of the function holds just as before. As for extractability, the extractor is defined exactly as before only that now it also has to output randomness r' for the homomorphic evaluation, which it will just sample uniformly at random. Now, even when $k \notin \mathrm{supp}(\mathcal{K}(1^n))$, extraction is still guaranteed. Indeed, the only case that changes is when σ is not sampled consistently with r and $i \neq 0^n$; here, both $\hat{\mathsf{c}}_{\mathcal{V},\mathcal{G}}$ corresponding to the extracted value, and $\hat{\mathsf{c}}_{\mathcal{G}}$ corresponding to the adversary's image would decrypt to 0.

We next show that oblivious key sampling holds. The image sampler $\mathcal{I}$, given a key $k = (\tau, \mathsf{c}_r)$ and an image $v = (y, \hat{\mathsf{c}}, i)$, acts according to two cases (similarly to the extractor). If $i = 0^n$, it can completely imitate the extractor choosing a canonical $u'_* = (i'; M', y', \pi'; s'; r')$ with $i' = n$, and outputting $\tilde{v} = f_k(u'_*)$. If $i \neq 0^n$, $\mathcal{I}$ outputs $\tilde{v} = (y, \hat{\mathsf{c}}'', i)$, where $\hat{\mathsf{c}}'' \leftarrow \mathsf{Eval}(\mathcal{C}_{\mathcal{G}}, \mathsf{c}_\tau)$ (and Eval is randomized). Indeed, the image output by $\mathcal{I}$ is the same as the one induced by the extractor, except that $\hat{\mathsf{c}}''$ may differ from the evaluated cipher $\hat{\mathsf{c}}'$ output by the extractor. However, as explained above, we are guaranteed that in this case $\mathsf{Dec}(\hat{\mathsf{c}}'') = \mathsf{Dec}(\hat{\mathsf{c}}')$, and so by the circuit privacy of the 1-Hop scheme, the output $\tilde{v}$ of $\mathcal{I}$ is computationally indistinguishable from the output v' induced by the extraction procedure Ext. (Recall that circuit privacy here holds even if the cipher c_r is sampled maliciously.) □

Remark 3.3 (one-wayness against superpolynomial adversaries). In Section 4.3.2, we shall require GEOWFs that are one way even against adversaries of size $\mathrm{poly}(T(n))$, for some superpolynomial function $T(n)$. Such GEOWFs can be obtained from our constructions, by using a PRG that is secure 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 [KRR], based on an appropriately strong private information retrieval scheme).

4 2-Message and 3-Message Zero-Knowledge against Verifiers with Bounded Advice

In this section, we define and construct two and three message ZK arguments against verifiers with bounded advice.

4.1 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 m, and using an arbitrary polynomial number of random coins.

Definition 4.1. *An argument system (P, V) is zero-knowledge against verifiers with m-bounded advice if for every PPT verifier V^*, there exists a PPT simulator* Sim *such that:*

$$\{\langle P(w) \leftrightarrows V^*(z)\rangle(x)\}_{\substack{(x,w)\in\mathcal{R}_\mathcal{L} \\ z\in\{0,1\}^{m(|x|)}}} \approx_c \{\mathsf{Sim}(z,x)\}_{\substack{(x,w)\in\mathcal{R}_\mathcal{L} \\ z\in\{0,1\}^{m(|x|)}}} ,$$

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

Remark 4.1 (universal simulator). In the above definition, each PPT V^* is required to have a designated PPT simulator Sim^*_V. Our constructions will, in fact, guarantee the existence of one universal simulator Sim that, in addition to (z, x), is also given the code of V^* and a bound $1^{t^*_V}$ on the running time of $V^*(x; z)$, and simulates V^*'s view. Moreover, the running time of Sim 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 m on the size of the advice.

4.2 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 $(x, w) \in \mathcal{R}_\mathcal{L}$ to be proven; in particular, it is of fixed polynomial length in the security parameter n, independently of $|x, w|$.

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 x. In Lapidot and Shamir's Hamiltonicity variant [LS90], the first message is independent of (x, w) themselves, but does depend on $|x, w|$ (see details in [OV12]). ZAPs do satisfy the independence requirement (as there is no first prover message at all), but they do not provide 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 (x, w), even of their length, and the second (verifier) message only depends on $|x|$.

Definition 4.2 (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 $x \in \mathcal{L} \cap \{0,1\}^\ell$, $w \in \mathcal{R}_\mathcal{L}(x)$, $n \in \mathbb{N}$:*

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

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 (x, w).

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

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

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

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

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

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

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

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 have 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 x, 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, BP13].

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 required that extraction is possible given any two valid transcipts $\vec{C}, \vec{C}'$ that share the same first message. Such an extractable commitment can be constructed from any perfectly-binding non-interactive commitment, see e.g. [PW09].

Lemma 4.1. *Protocol 1 is a 3-message WIPOK with instance-independent first message.*

We next prove the lemma. The proof is an adaptation of a proof from [BP13].

Proof. We start by showing that the protocol is WI. Let

$$(\mathcal{X}, \mathcal{W}_0, \mathcal{W}_1) = \{(x, w_0, w_1) : (x, w_0), (x, w_1) \in \mathcal{R}_{\mathcal{L}}\}$$

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 $w_0 \in \mathcal{W}_0$, and ending with an hybrid describing an interaction with a prover that uses $w_1 \in \mathcal{W}_1$, where both w_0, w_1, are witnesses for some $x \in \mathcal{X}$. 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

Protocol 1

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

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

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 $\ell = |x|$, 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 over $(r, C_0^{(2)}, C_1^{(2)}, C^{(2)})$.
3. P, given (x, 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 an ZAP proof π for the statement:

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

   - 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 1: A 3-message WIPOK with instance-independent first message

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

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.

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 $p_0 - p_1 \geq \epsilon$, for some noticeable $\epsilon = \epsilon(n)$. 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 $\frac{1+\epsilon}{2}$. 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 $\frac{p_0}{2} + \frac{1-p_1}{2} \geq \frac{1}{2} + \frac{\epsilon}{2}$. 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 $\epsilon = \epsilon(n)$, then with probability at least $\epsilon/2$ over its first message, it holds with probability at least $\epsilon/2$ 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 $\epsilon/2 - \mathrm{negl}(n)$ completes 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 4.1. □

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 replace with any 2-message WI with instance-independent first message.

4.3 Constructions

In this section, we construct zero-knowledge protocols against verifiers with bounded advice from generalized extractable one-way functions against adversaries with bounded advice (GEOWFs against BAPT adversaries). We start by describing a construction of a 3-message argument of knowledge from any GEOWF that is everywhere extractable and has oblivious image verification, and every 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 one-way against $\mathrm{poly}(T(n))$-size adversaries, and any 2-message WI with instance-independent verifier message (in particular, ZAPs).

4.3.1 A 3-message zero-knowledge argument of knowledge.

In what follows, let $\mathcal{F}$ be a family of GEOWFs, against m-BAPT adversaries, and assume that $\mathcal{F}$ is everywhere extractable and has an oblivious image sampler $\mathcal{I}$. We shall denote by $(\mathsf{wi}_1, \mathsf{wi}_2, \mathsf{wi}_3)$ the messages of a WIPOK with an instance-independent first message (as in Definition 4.2). The protocol is given in Figure 2.

Protocol 2

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

Auxiliary Input to P: a witness w for x.

1. P sends the first message $\mathsf{wi}_1 \in \{0,1\}^n$ of the instance-dependent WIPOK.
2. V samples $k \leftarrow \mathcal{K}(1^n; r), u \leftarrow \{0,1\}^{\ell(n)}$, computes $v = f_k(u)$, and sends (k, v) , as well as the second WIPOK message wi_2.
3. P samples $\tilde{v} \leftarrow \mathcal{I}(k, v)$, and sends $\tilde{v}$, together with the third WIPOK message wi_3 stating that:

$$\{x \in \mathcal{L}\} \bigvee \{\exists u : \ \tilde{v} = f_k(u) \ \} \ ,$$

   using the witness $w \in \mathcal{R}_{\mathcal{L}}(x)$.
4. V verifies the proof and tests that the equivalence relation holds by running $\mathcal{T}(v, \tilde{v}, r)$.

Figure 2: A 3-message ZK argument of knowledge against verifiers with m-bounded advice.

Theorem 4.1. *Protocol 2 is a zero-knowledge argument of knowledge against m-BAPT verifiers.*

Remark 4.2 (Instance-independent first-message). An additional feature of the protocol is that it preserves the first message instance-independence of the WIPOK system.

The high-level idea behind the proof is provided in the introduction, we now provide a more detailed proof sketch.

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

Claim 4.1. *Protocol 2 is an argument of knowledge against against arbitrary polysize provers.*

Proof sketch. Let P^* be any polysize prover that convinces V of accepting with noticeable probability $\epsilon = \epsilon(n)$. 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 (k, v) 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 $\mathsf{tr}_{\mathsf{wi}}$ 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 hopefuly obtains a witness for the WI statement

$$\{x \in \mathcal{L}\} \bigvee \{\exists u : \tilde{v} = f_k(u)\} \quad ,$$

where (k, v) are those honestly sampled by P^*_{wi}, and $\tilde{v}$ is output by P^*.

We claim that, with noticeable probability $\epsilon^2/2 - \mathrm{negl}(n)$, we find a witness w for the first part of the statement $x \in \mathcal{L}$. Otherwise, we can use P^* to break the (strong) one-wayness of $\mathcal{F}$. To prove the 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 ϵ. Now, let us call random coins r for P^*_{wi} good if they are such that with probability at least $\epsilon/2$ 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 ϵ, then by a standard averaging argument, at least an $\epsilon/2$ 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 $\epsilon \cdot \epsilon/2$. 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 $x \in \mathcal{L}$ statement. Indeed, assume that, with noticeable probability η , 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 u', and $v' = f_k(u')$, such that $\tilde{v} = f_k(u')$, where $\tilde{v}$ is the output of P^*. Moreover, since the transcript is accepting, we know that $v \sim v'$.

We can now construct an inverter that breaks the (strong) one-wayness of $\mathcal{F}$. The inverter, given (k, v) would simply emulate all of the experiment above on its own, where P_{wi} would use (k, v) to emulate the second verifier message, instead of sampling it on its own. By the above, it would obtain a (relative) preimage with noticeable probability η.

This completes the proof. □

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 $\widetilde{\mathsf{Sim}}$ 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 4.2. *Protocol 2 is ZK against any polytime verifier V^* with advice of size at most $m(n) - 2n$.*

Proof sketch. We describe a universal ZK simulator Sim and show its validity (universality is in the sense of Remark 4.1). Let $x \in \mathcal{L}$ and let V^* be the code of any malicious verifier, and let z' be any advice of length at most $m - 2n$. Sim 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 $V^*(x; z')$ who returns (k, v, wi_2) that are

(allegedly) a key for an extractable function, an image under the function, and the second message of the WIPOK.

Sim now constructs from the code of V^* a machine M_{V^*} that, given k and $z = (z', x, \mathsf{wi}_1)$ as input, outputs some v, and whose running time is linear in the running time t_{V^*} of V^*. Note that $|z| \leq |z'|+|x|+|\mathsf{wi}_1| \leq m(n)$, and thus Sim can apply the extractor Ext on M_{V^*}, and obtain $u' \in \{0,1\}^\ell$ in time $\mathrm{poly}(t^*_V)$. Sim now computes $v' = f_k(u')$ to V^*, and completes the WIPOK using the trapdoor u' as a witness.

The validity of the simulator now follows by witness indistinguishability, as well as the oblivious image sampling guarantee. Specifically, we can first move to a hybrid simulator Sim' that proves the WIPOK statement using the witness w. The view generated by Sim' is indistinguishable from the one generated by Sim due to the WI property. Now, we can claim that the view generated by Sim' is indistinguishable from that generated by honest prover P. Indeed, the only difference between the two is that P sends $\tilde{v} \leftarrow \mathcal{I}(k, v)$, whereas Sim' sends $v' = f_k(u')$, for the extracted input u'; however, by the oblivious image sampling guarantee $\tilde{v} \approx_c v'$. □

This completes the proof of Theorem 4.1. □

4.3.2 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.

In what follows, 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 $T(n) = n^{\omega(1)}$. Let $\mathcal{F}$ be a family of GEOWFs, against m-BAPT adversaries, and assume that $\mathcal{F}$ is everywhere extractable and has an oblivious image sampler $\mathcal{I}$. Further assume that $\mathcal{F}$ is one-way against adversaries of size $\mathrm{poly}(T)$ (see Remark 3.3). Also, we shall denote by $(\mathsf{wi}_1, \mathsf{wi}_2)$ the verifier and prover messages of a 2-message WI with an instance-independent first message (as in Definition 4.2).

Protocol 3

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

Auxiliary Input to P: a witness w for x.

1. V samples $k \leftarrow \mathcal{K}(1^n; r), u \leftarrow \{0,1\}^{\ell(n)}$, computes $v = f_k(u)$, and sends k, v , as well as the first WI message wi_1.
2. P samples a commitment to zero $C \leftarrow \mathcal{C}(0^\ell)$, and $\tilde{v} \leftarrow \mathcal{I}(k, v)$, and sends $C, \tilde{v}$, together with the second WI message wi_2 stating that:

$$\{x \in \mathcal{L}\} \bigvee \left\{ \exists u : \begin{array}{c} v = f_k(u) \\ C = \mathcal{C}(u) \end{array} \right\} ,$$

   using the witness $w \in \mathcal{R}_{\mathcal{L}}(x)$.
3. V verifies the proof and tests proximity by running $\mathcal{T}(v, \tilde{v}, r)$.

Figure 3: A 2-message ZK argument against verifiers with bounded advice.

Theorem 4.2. *Protocol 3 is a zero-knowledge argument against m-BAPT 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 we will construct an extractor that works 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 4.3. *Protocol 3 is an argument.*

Proof sketch. Let P^* be any polysize prover, and assume towards contradiction that for infinitely many $x \notin \mathcal{L}$, P^* convinces V of accepting with noticeable probability $\epsilon = \epsilon(n)$. We show to break the strong one-wayness of $\mathcal{F}$. The invereter, given (k, v) would sample a first WI message wi_1, and feed (k, v, wi_1) to P^*, who outputs a commitment C, an alleged image $\tilde{v}$, and a proof wi_2 for the statement

$$\{x \in \mathcal{L}\} \bigvee \left\{ \exists u : \begin{array}{l} \tilde{v} = f_k(u) \\ C = \mathcal{C}(u) \end{array} \right\} .$$

We know that with probability ϵ the proof is convincing, so by the soundness of the WI scheme, and since $x \notin \mathcal{L}$, it follows that C is a commitment to a (relative) preimage of v. The inverter can now break C in time $T(n)$ and thus break the strong one-wayness 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 4.4. *Protocol 3 is ZK against any polytime verifier V^* with advice of size at most $m(n) - n$.*

Proof sketch. We describe a universal ZK simulator Sim and show its validity (universality is in the sense of Remark 4.1). Let $x \in \mathcal{L}$ and let V^* be the code of any malicious verifier, and let z' be any advice of length at most $m - n$. Sim starts by running $V^*(x; z')$ who returns (k, v, wi_1) that are (allegedly) a key for an extractable function, an image of the of the function, and the verifier message of the WI protocol.

Sim now constructs from the code of V^* a machine M_{V^*} that, given k and $z = (z', x)$ as input, outputs some v, and whose running time is linear in the running time t_{V^*} of V^*. In particular, $|z| \leq |z'|+|x| \leq m(n)$. Sim then applies the extractor Ext on M_{V^*}, and obtains $u' \in \{0,1\}^\ell$ in time $\mathrm{poly}(t^*_V)$.

Sim now computes $v' = f_k(u')$, as well as a commitment C to u', and completes the WI using the trapdoor u' as a witness. It sends C, v', wi_2 to complete the simulation.

The validity of the simulator now follows by witness indistinguishability, as well as the oblivious image sampling guarantee. Specifically, we can first move to a hybrid simulator Sim' that proves the WIPOK statement using the witness w. The view generated by Sim' is indistinguishable from the one generated by Sim due to the WI property. Now, we can claim that the view generated by Sim' is indistinguishable from that generated by honest prover P. Indeed, the only difference between the two is that P commits to 0^ℓ instead of u', and sends $\tilde{v} \leftarrow \mathcal{I}(k, v)$, whereas Sim' sends $v' = f_k(u')$, for the extracted input u'. Thus, the two views are indistinguishable by the hiding of the commitment and by the oblivious image sampling guarantee that $\tilde{v} \approx_c v'$. □

This completes the proof of Theorem 4.2. □

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.

[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.

[BG08] Boaz Barak and Oded Goldreich. Universal arguments and their applications. *SIAM J. Comput.*, 38(5):1661–1694, 2008.

[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.

[BM84] Manuel Blum and Silvio Micali. How to generate cryptographically strong sequences of pseudo-random bits. *SIAM J. Comput.*, 13(4):850–864, 1984.

[BP04] 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.

[BP13] Nir Bitansky and Omer Paneth. On the impossibility of approximate obfuscation and applications to resettable cryptography. In *STOC*, pages 241–250, 2013.

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

[Can00] Ran Canetti. Security and composition of multiparty cryptographic protocols. *Journal of Cryptology*, pages 143–202, 2000.

[Can01] Ran Canetti. Universally composable security: A new paradigm for cryptographic protocols. In *Proceedings of the 42nd Annual IEEE Symposium on Foundations of Computer Science*, pages 136–145, 2001.

[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.

[CLP13] Kai-Min Chung, Huijia Lin, and Rafael Pass. Constant-round concurrent zero knowledge from p-certificates. In *FOCS*, 2013.

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

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

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

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

[FS89] Uriel Feige and Adi Shamir. Zero knowledge proofs of knowledge in two rounds. pages 526–544, 1989.

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

[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.

[GL89] O. Goldreich and L. A. Levin. A hard-core predicate for all one-way functions. In *STOC '89: Proceedings of the twenty-first annual ACM symposium on Theory of computing*, pages 25–32, New York, NY, USA, 1989. ACM.

[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.

[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.

[HILL99] Johan Håstad, Russell Impagliazzo, Leonid A. Levin, and Michael Luby. A pseudorandom generator from any one-way function. *SIAM J. Comput.*, 28(4):1364–1396, 1999.

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

[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.

[Kil92] Joe Kilian. A note on efficient zero-knowledge proofs and arguments. In *Proceedings of the 24th Annual ACM Symposium on Theory of Computing*, pages 723–732, 1992.

[KRR] Yael Tauman Kalai, Ran Raz, and Ron D. Rothblum. Delegation for p. In *Announcement at STOC13*.

[KRR13] Yael Tauman Kalai, Ran Raz, and Ron D. Rothblum. Delegation for bounded space. In *STOC*, pages 565–574, 2013.

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

[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.

[OI07] Rafail Ostrovsky and William E. Skeith III. A survey of single-database private information retrieval: Techniques and applications. In *Public Key Cryptography*, pages 393–411, 2007.

[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.

[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 BAPT 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 BAPT 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 m-BAPT adversaries where $m = n^{\Omega(1)}$, 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 Ext such that for any PPT adversary M, any large enough security parameter $n \in \mathbb{N}$, and any advice $z \in \{0,1\}^{m(n)}$:

$$\Pr_{k \leftarrow \mathcal{K}(1^n)} \left[\begin{array}{c} v \leftarrow M(k;z) \\ \exists u : f_k(u) = v \end{array} \wedge \begin{array}{c} u' \leftarrow \mathsf{Ext}^{M(\cdot;z)}(k) \\ f_k(u') \neq v \end{array} \right] \leq \mathrm{negl}(n) \ .$$

To see this, consider the adversary M that interprets its auxiliary input as a seed of a pseudo-random function PRF that maps the keys of $\mathcal{F}$ to inputs of $\mathcal{F}$. On input $(k; z)$, M computes an input $u = \mathsf{PRF}_z(k)$ and outputs $v = f_k(u)$. Using the guarantee of the pseudo-random function, it is not hard to see that any black-box extractor Ext can be used to break the one-wayness property of $\mathcal{F}$. Indeed, given (k, y), an inverter can simulate the view of Ext in an interaction with $M(\cdot; z)$ by answering any query $k' \neq k$ with a random image $f_{k'}(x)$, and answering the query k with y.

Note that the above does not hold when $m = O(\log(n))$, since then the advice cannot contain a seed for a secure pseudo-random function. In fact, when $m = O(\log(n))$, any family that is EOWF against m-BAPT adversaries also has a black-box extractor. The extractability property of the EOWF guarantees the existence of an extractor for every adversary M and advice z. Since there are only polynomially many different pairs (M, z), a black-box extractor can run the (possibly non-black-box) extractor for every such (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 BAPT 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 m-BAPT verifiers where $m = n^{\Omega(1)}$.

Similarly to the case EOWF, there is no impossibility for 3-message ZK against m-BAPT verifiers where $m = O(\log(n))$. In fact, as explained above, in this case, the non-black-box extractor of our EOWF 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 advice of the verifier can encode the first message of the protocol (and can in fact be extended to also rule out the case of BAPT 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 m-BAPT adversaries where m is smaller then the length of the first protocol message.