3-Message Zero Knowledge Against Human Ignorance

Nir Bitansky[1(✉)], Zvika Brakerski[2], Yael Kalai[3], Omer Paneth[4], and Vinod Vaikuntanathan[1]

[1] MIT, Cambridge, USA
nirbitan@csail.mit.edu
[2] Weizmann, Rehovot, Israel
[3] Microsoft Research, Cambridge, USA
[4] Boston University, Boston, USA

Abstract. The notion of Zero Knowledge has driven the field of cryptography since its conception over thirty years ago. It is well established that two-message zero-knowledge protocols for NP do not exist, and that four-message zero-knowledge arguments exist under the minimal assumption of one-way functions. Resolving the precise round complexity of zero-knowledge has been an outstanding open problem for far too long.

In this work, we present a three-message zero-knowledge argument system with soundness against uniform polynomial-time cheating provers. The main component in our construction is the recent delegation protocol for RAM computations (Kalai and Paneth, TCC 2016B and Brakerski, Holmgren and Kalai, ePrint 2016). Concretely, we rely on a three-message variant of their protocol based on a *key-less* collision-resistant hash functions secure against uniform adversaries as well as other standard primitives.

More generally, beyond uniform provers, our protocol provides a natural and meaningful security guarantee against real-world adversaries, which we formalize following Rogaway's "human-ignorance" approach (VIETCRYPT 2006): in a nutshell, we give an explicit uniform reduction from any adversary breaking the soundness of our protocol to finding collisions in the underlying hash function.

N. Bitansky—Research supported in part by DARPA Safeware Grant, NSF CAREER Award CNS-1350619, CNS-1413964 and by the NEC Corporation.

Z. Brakerski—Supported by the Israel Science Foundation (Grant No. 468/14), the Alon Young Faculty Fellowship, Binational Science Foundation (Grant No. 712307) and Google Faculty Research Award.

O. Paneth—Supported by the Simons award for graduate students in Theoretical Computer Science and an NSF Algorithmic foundations grant 1218461.

V. Vaikuntanathan—Research supported in part by DARPA Grant number FA8750-11-2-0225, NSF CAREER Award CNS-1350619, NSF Grant CNS-1413964 (MACS: A Modular Approach to Computer Security), Alfred P. Sloan Research Fellowship, Microsoft Faculty Fellowship, NEC Corporation and a Steven and Renee Finn Career Development Chair from MIT.

M. Hirt and A. Smith (Eds.): TCC 2016-B, Part I, LNCS 9985, pp. 57–83, 2016.
DOI: 10.1007/978-3-662-53641-4_3

1 Introduction

The fascinating notion of zero knowledge, conceived over thirty years ago by Goldwasser, Micali and Rackoff [GMR89], has been the source of a great many ideas that revolutionized cryptography, including the simulation paradigm and passive-to-active security transformations [GMW91, FLS99, Bar01, IKOS09].

A central and persistent open question in the theory of zero knowledge is that of round complexity (also called message complexity), which refers to the number of messages that the prover and the verifier must exchange in a zero-knowledge protocol. The seminal work of Goldreich, Micali and Wigderson [GMW91] showed the first computational zero-knowledge *proof* system for all of NP. Their protocol required a polynomial (in the security parameter) number of rounds (in order to achieve an exponentially small soundness error). Feige and Shamir [FS89] show a *four-round* computational zero-knowledge *argument* system [BCC88] for all of NPbased on algebraic assumptions.[1] The assumption was reduced to the minimal assumption of one-way functions by Bellare, Jakobsson and Yung [BJY97].

In terms of lower bounds, Goldreich and Oren [GO94] showed that *three rounds* are necessary for non-trivial zero knowledge (arguments as well as proofs) against non-uniform adversarial verifiers. Zero knowledge in the presence of verifiers with non-uniform advice has by now become the gold standard as it is often essential for secure composition (see, e.g., [GK96b]).

This state of affairs leaves behind a question that has been open for far too long:

What is the minimal round-complexity of zero knowledge?

By the works of [FS89, BJY97], the answer is at most 4 while Goldreich and Oren tell us that the answer is at least 3. So far, all constructions of three-message computational zero-knowledge argument systems for NPwere based on strong "auxiliary-input knowledge assumptions" [HT98, BP04b, CD09, BP12, BCC+14]. The plausibility of these assumptions was questioned already around their introduction [HT98] and they were recently shown to be false assuming the existence of *indistinguishability obfuscation* [BCPR14, BM14]. In summary, finding a three-message zero-knowledge argument (under reasonable, falsifiable assumptions) matching the Goldreich-Oren lower bound remains wide open.

Why is Three-Message Zero Knowledge so Interesting. Aside from its significance to the theory of zero knowledge, the question of three-message zero knowledge is also motivated by its connections to two fundamental notions in cryptography, namely *non-black-box security proofs* and *verifiable computation*.

In order to make sense of this, let us tell you the one other piece of the zero knowledge story. An important dimension of zero-knowledge proofs is whether the zero-knowledge simulator treats the (adversarial) verifier as a black-box or

[1] While zero-knowledge *proofs* [GMW91] provide soundness against computationally unbounded cheating provers, zero-knowledge *arguments* [BCC88] are weaker in that they provide soundness only against computationally bounded cheating provers.

not. In all the protocols referenced above (with the exception of the ones based on "auxiliary-input knowledge assumptions"), the simulator treats the verifier as a black box. Goldreich and Krawczyk [GK96b] show that in any three-message zero-knowledge protocol for a language outside BPP, the simulator *must* make non-black-box use of the verifier's code. In other words, any future three-message zero-knowledge protocol has to "look different" from the ones referenced above.

The pioneering work of Barak [Bar01] demonstrated that barriers of this kind can sometimes be circumvented via non-black-box simulation. However, Barak's technique, and all other non-black-box techniques developed thus far, have only led to protocols with at least four messages [BP13, COP+14].

A bottleneck to reducing the round-complexity of Barak's protocol is the reliance on four-message *universal arguments* [BG08], a notion that enables fast verification of NPcomputations. Accordingly, developments in round-efficient systems for verifiable computation may very well lead to corresponding developments in three-message zero knowledge. In fact, strong forms of verifiable computation have recently proven instrumental in producing novel non-black-box simulation techniques, such as in the context of constant-round concurrent protocols [CLP13b, CLP15]. It is natural, then, to wonder whether these and related developments help us construct three-message zero-knowledge argument systems.

On Uniform (and Bounded Non-uniform) Verifiers. Bitansky, Canetti, Paneth and Rosen [BCPR14] study three-message protocols satisfying a relaxed notion of zero knowledge. Instead of requiring the zero knowledge guarantee against all non-uniform verifiers, they only consider verifiers that have an a-priori bounded amount of non-uniformity (but may still run for an arbitrary polynomial time). This includes, in particular, zero-knowledge against uniform verifiers. They demonstrate a three-message zero-knowledge protocol against verifiers with bounded non-uniformity based on the verifiable delegation protocol of Kalai, Raz, and Rothblum [KRR14].

Notably, restricting attention to verifiers with bounded uniformity comes with a great compromise. For once, the zero knowledge property is not preserved under sequential composition. More broadly, such protocols may not provide a meaningful security guarantee against real-world adversaries. As a concrete example, the zero knowledge property of the protocol in [BCPR13] crucially relies on the fact that messages sent by the verifier can be simulated by a Turing machine with a short description, shorter than the protocol's communication. However, this assumption may not hold for real-world adversaries, which can certainly have access to arbitrarily long strings with no apparent short description.

1.1 Our Results

In this work, we construct a three-message argument for NPthat is zero knowledge against fully non-uniform verifiers and sound against provers with a-priori bounded (polynomial amount of) non-uniformity. The main component in our

construction is a verifiable delegation protocol for RAM computations recently constructed by Kalai and Paneth [KP15] and improved by Brakerski, Holmgren and Kalai [BHK16]. Concretely, we rely on a three-message variant of the [BHK16] protocol based on *keyless* collision-resistant hash functions secure against adversaries with bounded non-uniformity and slightly super-polynomial running time, and a (polynomially-secure) computational private information retrieval (PIR) scheme, as well as other more standard cryptographic assumptions.

In contrast to the setting of verifiers with bounded non-uniformity, our protocol remains secure under sequential composition. Furthermore, our protocol provides a natural and meaningful security guarantee against real-world adversaries, which we formalize following Rogaway's "human ignorance approach" [Rog06], described in greater detail below.

Rogaway's "Human Ignorance" Approach and Real-World Security. A more informative way of describing the soundness of our protocol is by the corresponding security reduction. We construct a zero-knowledge *argument* system, meaning that the soundness of the protocol is computational. That is, any prover that breaks the soundness of our protocol, regardless of how non-uniform it is, can be *uniformly* turned into a collision finder for an underlying hash function. In other words, there is a *uniform* algorithm called collision-finder who finds collisions in the hash function given oracle access to the soundness-breaker.

In our protocol, the hash function must already be determined before the first message is sent, thus requiring that we rely on a fixed (key-less) function as opposed to a function family as is normally the case when dealing with collision-resistant hash functions. Clearly, a fixed hash function cannot be collision-resistant against non-uniform adversaries (as such an adversary can have a collision for the function hard-wired as part of its non-uniform advice). However, as argued by Rogaway, a *uniform* reduction from finding collisions in such a function to breaking the security of a protocol is sufficient to argue the real-world security of the protocol. Briefly, the rationale is that an adversarial algorithm that breaks the security of the protocol (with or without non-uniform advice) can be turned into an explicit algorithm that finds collisions in the hash function (with the *same* non-uniform advice). Indeed, for common constructions of hash functions, such as SHA-3, collisions (while they surely exist) are simply not known.

Our main result can accordingly be stated as follows.

Informal Theorem 1.1 [See Theorem 3.1]**.** *Assuming the existence of a computational private information retrieval (PIR) scheme, a circuit-private 1-hop homomorphic encryption scheme, and a non-interactive commitment scheme, there exists a three-message argument for* NP *with a uniform reduction* $\mathcal{R}$ *(described in the proof of Theorem 3.1) running in quasi-polynomial time, such that, for every non-uniform PPTadversary* $\mathcal{A}$*, if* $\mathcal{A}$ *breaks the soundness of the protocol instantiated with a keyless hash function* $\mathcal{H}$*, then* $\mathcal{R}^{\mathcal{A}}$ *outputs a collision in* $\mathcal{H}$*. The protocol is zero knowledge against non-uniform probabilistic polynomial-time (PPT) verifiers.*

All the cryptographic primitives (except the key-less hash function) can be instantiated from the learning with errors (LWE) assumption [Reg09].

Asymptotic Interpretations. As discussed above, implementing our protocol with a key-less hash function such as SHA-3 guarantees security against "ignorant" adversaries that are unable to find hash collisions. This class of adversaries may include all the adversaries we care about in practice, however, since functions like SHA-3 do not provide any asymptotic security, we cannot use standard asymptotic terminology to define the class of "SHA-ignorant adversaries".

We formalize the security of our protocol and hash function in conventional asymptotic terms. For any asymptotic hash family $\mathcal{H} = \{\mathcal{H}_n\}_{n\in\mathbb{N}}$, we can accordingly think of the class of adversaries that are $\mathcal{H}$-ignorant. Trying to capture more natural classes of adversaries, we focus on the subclass of adversaries with bounded non-uniformity. It may be reasonable to assume that an asymptotic keyless hash function is indeed collision-resistant against this class as long as the corresponding non-uniform advice is shorter than the hash input length. Therefore, the result for adversaries with bounded non-uniformity stated above follows as a corollary of our explicit reduction.

The Global Common Random String Model and Resettable Security. Another direct corollary of our result is that assuming (the standard notion of) keyed collision-resistant hash-function families, there is a 3-message zero-knowledge protocol that is sound against fully non-uniform provers in the *global (or non-programable) common random string model* [Pas03, CDPW07] or in the global hash model [CLP13a]. As observed in [Pas03], both the Goldreich-Oren lower bound and the Goldreich-Krawczyk black-box lower bound hold even in these models.

Another property of our protocol is that it can be made resettably sound [BGGL01] via the (round-preserving) transformation of Barak, Goldreich, Goldwasser and Lindell [BGGL01]. This holds for the three-message version of the protocol (against provers with bounded uniformity, or alternatively, against non-uniform provers in the global random string model).

1.2 Our Techniques

We now give an overview of the main ideas behind the new protocol.

Barak's Protocol. As explained above, three-message zero-knowledge can only be achieved via *non-black-box* simulation (and the Goldreich-Krawczyk lower bound, in fact, holds even when considering uniform provers). Thus, a natural starting point is the non-black-box simulation technique of Barak [Bar01], which we outline next. Following the Feige-Lapidot-Shamir paradigm [FLS99], the prover and verifier in Barak's protocol first execute a *trapdoor generation preamble*: the verifier sends a key h for a collision-resistant hash function, the prover responds with a commitment cmt, and then, the verifier sends a random challenge u. The preamble defines a "trapdoor statement" asserting that there exists a program Π such that cmt is a commitment to $h(\Pi)$ and $\Pi(\mathsf{cmt})$ outputs

u. Intuitively, no cheating prover is able to commit to a code that predicts the random u ahead of time, and thus cannot obtain a witness (a program Π) for the trapdoor statement. In contrast, a simulator that is given the code of the (malicious) verifier, can commit to it in the preamble and use it as the witness for the trapdoor statement.

In the second stage of the protocol, the prover and the verifier engage in a witness-indistinguishable (WI) protocol intended to convince the verifier that either the real statement or the trapdoor statement is true, without revealing to the verifier which is the case. Here, since the trapdoor statement corresponds to a computation $\Pi(\mathsf{cmt})$ that may be longer than the honest verifier's runtime, a standard WI system is insufficient. This difficulty is circumvented using the 4-message universal arguments mentioned before, where verification time is independent of the statement being proven.

Overall, Barak's protocol is executed in six messages. In the first message, the verifier sends a key for a collision-resistant hash function, which effectively serves both as the first message (out of three) of the preamble and as the first message (out of four) of the universal argument to come. Then, the two remaining messages of the preamble are sent, following by the remaining three messages of a WI universal argument.[2]

Squashing Barak's Protocol. To achieve a three-message protocol, we will squash Barak's protocol. Using a keyless hash function, we can eliminate the first verifier message (which, in Barak's protocol, consists of a key for a collision-resistant hash function). It is just this step that restricts our soundness guarantee to only hold against provers that are unable to find collisions in the key-less hash function (e.g., provers with bounded non-uniformity). This leaves us with a *five*-message protocol, which is still worse than what is achievable using black-box techniques. The bulk of the technical contribution of this work is devoted to the task of squashing this protocol into only *three* messages.

Having eliminated the verifier's first message, we are now left with a 2-message preamble followed by a 3-message WI universal argument. A natural next step is to attempt executing the preamble and the WI argument in parallel. The main problem with this idea is that in Barak and Goldreich's universal arguments, the statement must be fixed before the first prover message is computed. However, in the protocol described, the trapdoor statement is only fixed once the entire preamble has been executed.

We observe that, paradoxically, while the trapdoor statement is only fixed after the preamble has been executed, *the witness for this statement is fixed before the protocol even starts*! Indeed, the witness for the trapdoor statement is simply the verifier's code. It is therefore sufficient to replace Barak and Goldreich's universal argument with a 3-message verifiable delegation protocol that has the following structure: the first prover message depends on the witness alone, the verifier's message fixes the statement, and the third and last prover response includes the proof (which already depends on both the statement and witness).

[2] Barak's original construction, in fact, consists of seven messages, but can be squashed into six by using an appropriate WI system (see, e.g., [OV12]).

Verifiable Memory Delegation. To obtain a verifiable delegation scheme with the desired structure, we consider the notion of verifiable memory delegation [CKLR11]. In memory delegation, the prover and verifier interact in two phases. In the offline phase, the verifier sends a large memory string m to the prover, saving only a short digest of m. In the online phase, the verifier sends a function f to the prover and the prover responds with the output $f(m)$ together with a proof of correctness. The time to verify the proof is independent of the memory size and the function running time.

In our setting, we think of the memory as the witness and of the delegated function as verifying that its input is a valid witness for a specified statement (encoded in the function). One important difference between the settings of verifiable memory delegation and ours is that in the former, the offline phase is executed by the verifier, but in our setting, the prover may adversarially choose any digest (which may not even correspond to any memory string). We therefore rely on memory delegation schemes that remain secure for an adversarially chosen digest. We observe that the verifiable delegation protocols for RAM computations of [KP15, BHK16] yield exactly such a memory delegation scheme, and when implemented using a keyless hash function this delegation scheme is secure against the class of adversaries that cannot find collisions in the hash function (e.g. adversaries with bounded non-uniformity).

Fulfilling the above plan encounters additional hurdles. The main such hurdle is the fact that the verifiable delegation schemes of [KP15, BHK16] are not witness-indistinguishable. We ensure witness-indistinguishability by leveraging special properties of the Lapidot-Shamir WI protocol [LS90a, OV12], and 1-hop homomorphic encryption [GHV10] (similar ideas were used in [BCPR14]).

1.3 More Related Work

We mention other related works on round-efficient zero knowledge.

On Zero-Knowledge Proof Systems. In this work we show a 3-message *argument* system for NP. If one requires a *proof* system instead, with soundness against unbounded provers, Goldreich and Kahan [GK96a] showed a 5-round (black-box) zero-knowledge proof system for NP. On the other hand, Katz [Kat12], extending the result of Goldreich and Krawczyk [GK96b], shows that, assuming the polynomial hierarchy does not collapse, zero-knowledge protocols for an NP-complete language require at least 5 rounds if the simulator only makes black-box use of the verifier's code. The question of 3-round and 4-round zero-knowledge proof systems for NP(necessarily with non-black-box simulation) still remains wide open.

On Quasi-Polynomial Time Simulation. Barak and Pass [BP04a] show a 1-round *weak* zero-knowledge argument for NPwith soundness against uniform polynomial-time provers, based on non-standard assumptions. (One of their assumptions is the existence of a key-less collision-resistant hash function against uniform adversaries with sub-exponential running time.) Their notion of *weak* zero knowledge allows for a quasi-polynomial-time simulator. The fact that the

simulator can run longer than (any possible) cheating prover means that the simulator can (and does) break the soundness of the protocol. This has the effect that the round-complexity lower bounds referenced above do not apply in this model. Furthermore, such a protocol may leak information that cannot be simulated in polynomial time (but only in quasi-polynomial time).

Organization. In Sect. 2, we give the basic definitions used throughout the paper, including the modeling of adversaries and reductions, the definition of keyless hash functions, and memory delegation. In Sect. 3, we describe and analyze the new protocol.

2 Definitions and Tools

In this section, we define the adversarial model we work in, zero-knowledge protocols against restricted classes of provers (e.g., ones with bounded non-uniformity), as well as the tools used in our construction.

2.1 Modeling Adversaries, Reductions, and Non-uniformity

In this section, we recall the notion of (black-box) reductions, and address two general classes of adversaries touched in this paper. Commonly in crypto, we consider (uniform) polynomial time reductions between different non-uniform polynomial-time adversaries. In this paper, we will sometimes consider more general types of reductions, e.g. uniform reductions that run in slightly super-polynomial time, as well as different classes of adversaries, e.g. uniform PPT adversaries, or adversaries with bounded non-uniformity. In such cases, we will be explicit about the concrete classes of reductions and adversaries involved.

Rogaway's "Human Ignorance" Approach to Reductions. As discussed in the introduction, the most informative way of describing the soundness of our protocol is by the corresponding security reduction from collision-resistance to soundness. Rogaway [Rog06] suggests a framework for formalizing such statements. In this work, however, for the sake of simpler exposition, we do not fully follow Rogaway's framework; we explain the differences next.

While Rogaway's approach gives a meaningful result even for non-asymptotic hash functions such as SHA-3 in terms of concrete security, our security definitions are still formalized in asymptotic terms. We parameterize the security definitions by the class of adversaries. Our main theorem states that for every class of adversaries $\mathbb{A}$, the soundness of the protocol against adversaries in $\mathbb{A}$ can be reduced to the security of the hash function against the same class of adversaries.

We note that the security of our protocol is based on other primitives except keyless collision-resistant hash. In our theorems, we do not emphasize the reduction to these primitives; rather, we simply restrict our result only to classes of adversaries that are unable to break the security of these primitives (most naturally non-uniform polynomial time adversaries).

Reductions. For two classes of adversaries $\mathbb{R}, \mathbb{A}$, we denote by $\mathbb{R}^{\mathbb{A}}$ the class of adversaries $\mathcal{R}^{\mathcal{A}} = \left\{\mathcal{R}_n^{\mathcal{A}_n}\right\}_{n\in\mathbb{N}}$ where $\mathcal{R}_n$ makes calls to $\mathcal{A}_n$.[3]

The class $\mathbb{P}$ of non-uniform PPT adversaries. A general class of adversaries considered in this paper are non-uniform probabilistic polynomial-time Turing machines, or in short non-uniform PPT, which we denote by $\mathbb{P}$. Any such adversary $\mathcal{A} \in \mathbb{P}$ is modeled as a sequence $\{\mathcal{A}_n\}_{n\in\mathbb{N}}$, where n is the security parameter, and where the description and running time of $\mathcal{A}_n$ are polynomially bounded in n.

For a super-polynomial $\gamma(n) = n^{\omega(1)}$, we denote by $\mathbb{P}_\gamma$ the class of non-uniform probabilistic adversaries whose description and running time are polynomial in $\gamma(n)$.

The class $\mathbb{B}$ of PPT adversaries with bounded non-uniformity. We shall also consider the class $\mathbb{B}_\beta \subset \mathbb{P}$ of adversaries with bounded non-uniformity $O(\beta)$. Concretely, for a fixed function $\beta(n) \leq n^{O(1)}$, the class $\mathbb{B}_\beta$ consists of all non-uniform adversaries $\mathcal{A} \in \mathbb{P}$ whose description $|\mathcal{A}_n|$ is bounded by $O(\beta(n))$, *but their running time could be an arbitrary polynomial.* Abusing notation, we denote by $\mathbb{B}_0$ the class of *uniform* PPT *adversaries.*

For a super-polynomial function $\gamma(n) = n^{\omega(1)}$, we denote by $\mathbb{B}_{\beta,\gamma}$ the class of non-uniform probabilistic adversaries whose description is bounded by $O(\beta(n))$ (or the class of uniform probabilistic adversaries if $\beta = 0$) and running time is polynomial in $\gamma(n)$.

2.2 Zero Knowledge Arguments of Knowledge Against Provers with Bounded Non-uniformity

The standard definition of zero knowledge [GMR89, Gol04] considers general non-uniform provers (and verifiers). We define soundness (or argument of knowledge) more generally against provers from a given class $\mathbb{A} \subset \mathbb{P}$. In particular, we will be interested in strict subclasses of $\mathbb{P}$, such as adversaries with bounded non-uniformity.

In what follows, we denote by $\langle P \leftrightarrows V\rangle$ a protocol between two parties P and V. For input w for P, and common input x, we denote by $\langle P(w) \leftrightarrows V\rangle(x)$ the output of V in the protocol. For honest verifiers this output will be a single bit indicating acceptance (or rejection), whereas we assume (without loss of generality) that malicious verifiers outputs their entire view. Throughout, we assume that honest parties in all protocols are uniform PPT algorithms.

Definition 2.1. *A protocol $\langle P \leftrightarrows V\rangle$ for an* NP *relation $\mathcal{R}_\mathcal{L}(x, w)$ is a zero knowledge argument of knowledge against provers in class $\mathbb{A} \subset \mathbb{P}$ if it satisfies:*

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

$$\Pr\left[\langle P(w) \leftrightarrows V\rangle(x) = 1\right] = 1.$$

[3] In this paper, we shall explicitly address different classes of black-box reductions. One can analogously define non-black-box reductions.

2. **Computational zero knowledge:** *For every non-uniform PPT verifier* $V^* = \{V_n^*\}_{n\in\mathbb{N}} \in \mathbb{P}$*, there exists a (uniform) PPT simulator* $\mathcal{S}$ *such that:*

$$\{\langle P(w) \leftrightarrows V_n^*(x)\rangle\}_{\substack{(x,w)\in\mathcal{R}_\mathcal{L}\\ |x|=n}} \approx_c \{\mathcal{S}(V_n^*, x)\}_{\substack{(x,w)\in\mathcal{R}_\mathcal{L}\\ |x|=n}}.$$

3. **Argument of knowledge:** *There is a uniform PPT extractor* $\mathcal{E}$*, such that for any noticeable function* $\varepsilon(n) = n^{-O(1)}$*, any prover* $P^* = \{P_n^*\}_{n\in\mathbb{N}} \in \mathbb{A}$*, any security parameter* $n \in \mathbb{N}$*, and any* $x \in \{0,1\}^n$ *generated by* P_n^* *prior to the interaction:*

$$\text{if } \Pr\left[\langle P_n^* \leftrightarrows V\rangle(x) = 1\right] \geq \varepsilon(n),$$
$$\text{then } \Pr\left[\begin{matrix} w \leftarrow \mathcal{E}^{P_n^*}(1^{1/\varepsilon(n)}, x) \\ w \notin \mathcal{R}_\mathcal{L}(x)\end{matrix}\right] = \text{negl}(n).$$

2.3 Collision-Resistant Hashing

We define the notion of a keyless hash function that is collision resistant against a class $\mathbb{A} \subseteq \mathbb{P}_\gamma$ of adversaries. In particular, the definition may be realizable only for strict subclasses of $\mathbb{P}_\gamma$, such as the class $\mathbb{B}_{\beta,\gamma}$ of adversaries with bounded non-uniformity and poly$(\gamma(n))$ running time (where the description length of the adversary, namely β, will be shorter than the length of the input to the hash).

Definition 2.2. *Let* $n < \ell(n) \leq n^{O(1)}$*. A polynomial-time computable function*

$$\mathcal{H} = \{\mathcal{H}_n\}_{n\in\mathbb{N}}, \mathcal{H}_n : \{0,1\}^{\ell(n)} \to \{0,1\}^n,$$

is collision resistant against adversaries in $\mathbb{A}$ *if for any* $\mathcal{A} = \{\mathcal{A}_n\}_{n\in\mathbb{N}} \in \mathbb{A}$*, and every* $n \in \mathbb{N}$

$$\Pr\left[\begin{matrix} x, y \leftarrow \mathcal{A}_n; \\ \mathcal{H}_n(x) = \mathcal{H}_n(y)\end{matrix}\right] = \text{negl}(n).$$

where the probability is over the coins of $\mathcal{A}_n$.

Instantiation. Common constructions of keyless hash functions such as SHA-3 have a fixed output length and therefore do not directly provide a candidate for an asymptotic hash function as in Definition 2.2. One way to obtain candidates for an asymptotic hash function is to start with a family $\mathcal{H}'$ of (keyed) hash-functions

$$\mathcal{H}' = \left\{\mathcal{H}'_{n,k}\right\}_{n\in\mathbb{N},k\in\{0,1\}^n}, \mathcal{H}'_{n,k} : \{0,1\}^{\ell(n)} \to \{0,1\}^n,$$

and fix a uniform polynomial time algorithm K that given a security parameter 1^n outputs a key $k \in \{0,1\}^n$. The keyless hash $\mathcal{H}$ is then given by

$$\mathcal{H}_n = \mathcal{H}'_{n,K(1^n)}.$$

For $\mathcal{H}_n$ to be a good candidate collision resistant hash against adversaries in $\mathbb{B}_\beta$, we should make sure that $\beta = o(\ell)$, the family $\mathcal{H}'$ is collision resistant, and the algorithm K behaves "sufficiently like a random oracle". For example we can choose an algorithm K that uses a hash function like SHA-3 (or a version of it that can hash strings of arbitrary length) as a random oracle to output sufficiently many random bits.

2.4 Memory Delegation with Public Digest

A two-message memory delegation scheme [CKLR11] allows a client to delegate a large memory to an untrusted server, saving only a short digest of the memory. The client then selects a deterministic computation to be executed over the memory and delegates the computation to the server. The server responds with the computation's output as well as a short proof of correctness that can be verified by the client in time that is independent of that of the delegated computation and the size of the memory.

The notion of memory delegation we consider differs from that of [CKLR11] in the following ways.

- **Read-only computation.** We do not consider computations that update the memory. In particular, the digest of the delegated memory is computed once and does not change as a result of the computations.
- **Soundness.** We define soundness more generally for servers from a given class $\mathbb{A} \subset \mathbb{P}$. Whereas soundness is usually required against the class of all non-uniform PPT adversaries $\mathbb{P}$, we will also be interested in strict subclasses of $\mathbb{P}$, such as adversaries with bounded non-uniformity.
- **Soundness for slightly super-polynomial computations.** We require soundness to hold even for delegated computations running in slightly super-polynomial time.
- **Public digest.** We require that the digest of the memory can be computed non-interactively, and can be made public and used by any client to delegate computations over the same memory without compromising soundness. In particular, the client is not required to save any secret state when delegating the memory.
  Importantly, we do not assume that the party computing the digest is honest. We require that no efficient adversary can produce valid proofs for two different outputs for the same computation with respect to the same digest, even if the digest and computation are adversarially chosen.[4]
- **First message independent of function being delegated.** The first message of the delegation scheme (denoted below by q) depends only on the security parameter, and does not depend on the public digest or on the function being delegated.

Concretely, a two-message memory delegation scheme with public digest consists of four polynomial-time algorithms:

- $d \leftarrow \mathsf{Digest}(1^n, D)$ is a deterministic algorithm that takes a security parameter 1^n and memory D and outputs a digest $d \in \{0,1\}^n$.
- $(q, \tau) \leftarrow \mathsf{Query}(1^n)$ is a probabilistic algorithm that outputs a query q and a secret state τ. We assume w.l.o.g that the secret state τ is simply the random coins used by Query.

[4] Soundness with respect to an adversarial digest can be defined in a stronger way, for example, requiring knowledge of the memory corresponding to the digest. However, this stronger requirement is not necessary for our application.

- $\pi \leftarrow \mathsf{Prov}(1^t, \mathcal{M}, D, q)$ is a deterministic algorithm that takes a description of a Turing machine $\mathcal{M}$ and a bound t on the running time of $\mathcal{M}(D)$ and outputs a proof $\pi \in \{0,1\}^n$.
- $\mathsf{b} \leftarrow \mathsf{Ver}(d, \tau, \mathcal{M}, t, y, \pi)$ is a deterministic algorithm that takes a computation output y and outputs an acceptance bit b.

Definition 2.3 (Memory Delegation with Public Digest). *Let $\gamma(n)$ be a super-polynomial function such that $n^{\omega(1)} = \gamma(n) < 2^n$. A two-message memory delegation scheme* $(\mathsf{Digest}, \mathsf{Query}, \mathsf{Prov}, \mathsf{Ver})$ *for γ-time computations with public digest against provers in a class $\mathbb{A} \subset \mathbb{P}$ satisfies the following.*

- **Completeness.** *For every security parameter $n \in \mathbb{N}$, every Turing machine $\mathcal{M}$ and every memory $D \in \{0,1\}^*$ such that $\mathcal{M}(D)$ outputs y within $t \leq 2^n$ steps:*

$$\Pr\left[1 = \mathsf{Ver}(d, \tau, \mathcal{M}, t, y, \pi) \;\middle|\; \begin{array}{r} d \leftarrow \mathsf{Digest}(1^n, D) \\ (q, \tau) \leftarrow \mathsf{Query}(1^n) \\ \pi \leftarrow \mathsf{Prov}(1^t, \mathcal{M}, D, q) \end{array}\right] = 1.$$

- **Soundness.** *For every adversary $\mathcal{A} = \{\mathcal{A}_n\}_{n\in\mathbb{N}} \in \mathbb{A}$, there exists a negligible function* $\mathrm{negl}(\cdot)$ *such that for every security parameter $n \in \mathbb{N}$,*

$$\Pr\left[\begin{array}{l} t \leq \gamma(n) \\ y \neq y' \\ 1 = \mathsf{Ver}(d, \tau, \mathcal{M}, t, y, \pi) \\ 1 = \mathsf{Ver}(d, \tau, \mathcal{M}, t, y', \pi') \end{array} \;\middle|\; \begin{array}{l} (\mathcal{M}, t, d, y, y') \leftarrow \mathcal{A}_n \\ (q, \tau) \leftarrow \mathsf{Query}(1^n) \\ (\pi, \pi') \leftarrow \mathcal{A}_n(q) \end{array}\right] = \mathrm{negl}(n).$$

Instantiation. A memory delegation scheme satisfying Definition 2.3 can be obtained based on the delegation schemes for RAM computations of Kalai and Paneth [KP15] and that of Brakerski, Holmgren and Kalai [BHK16] with slight adaptations.[5] Below we describe the required adaptations. We focus on the scheme of [BHK16] that can be instantiated based on polynomially-secure PIR.

- **Remove public parameters.** The scheme of [BHK16] has public parameters that are generated honestly before the memory is delegated. These parameters consist of the description of a hash function chosen randomly from a family of collision-resistant hash functions. Here we remove the public parameters and instead use a keyless collision resistant hash against adversaries from a restricted class $\mathbb{A}$. (E.g., $\mathbb{A}$ can be the class of adversaries with β-bounded non-uniformity $\mathbb{B}_\beta$.) The security of our modified scheme against provers from $\mathbb{A}$ follows the same argument as in [BHK16], who show a uniform black-box reduction from a cheating prover to an adversary that finds collisions.
- **Soundness for slightly super-polynomial computations.** While the scheme of [BHK16] has completeness even for exponentially long delegated

[5] We note that we cannot use here the memory delegation scheme of [CKLR11,KRR14] since the soundness of their scheme assumes that the digest is honestly generated.

computations, soundness is only proved when the delegated computation is polynomial time. Here we require soundness even against slightly super-polynomial time $\gamma = n^{\omega(1)}$. In the [BHK16] reduction the running time of the adversary breaking the hash is proportional to the running time of the delegated computation. Therefore, soundness for slightly super-polynomial computations follows by the same argument, assuming a slightly stronger collision-resistance against adversaries from $\mathbb{B}^{\mathbb{A}}_{0,\gamma}$ who can run in time γ and use $\mathbb{A}$ as a black box.

Recall that $\mathbb{B}^{\mathbb{A}}_{0,\gamma}$ is the class of uniform probabilistic machines running in time $\gamma(n)^{O(1)}$ and given oracle access to an adversary in $\mathbb{A}$. Brakerski, Holmgren and Kalai prove that there is a $\gamma(n)^{O(1)}$-time uniform reduction from breaking the soundness of their scheme to breaking any underlying hash function, assuming the existence of a (polynomially secure) computational PIR scheme.

Theorem 2.1 [BHK16]. *For any $\mathbb{A} \subset \mathbb{P}$ and (possibly super-polynomial) function $\gamma(\cdot)$, assuming collision-resistant hash functions against adversaries in $\mathbb{B}^{\mathbb{A}}_{0,\gamma}$ and a computational PIR scheme, there exists a two-message memory delegation scheme for γ-time computations with public digest against provers in $\mathbb{A}$.*

2.5 Witness Indistinguishability with First-Message-Dependent Instances

We define 3-message WI proofs of knowledge where the choice of statement and witness may depend on the first message in the protocol. In particular, the first message is generated independently of the statement and witness. Also, while we do allow the content of the message to depend on the length ℓ of the statement, the message length should be of fixed to n (this allows to also deal with statements of length $\ell > n$). The former requirement was formulated in several previous works (see, e.g., [HV16]) and the latter requirement was defined in [BCPR14].

Definition 2.4 (WIPOK with first-message-dependent instances). *Let $\langle P \leftrightarrows V \rangle$ be a 3-message argument for $\mathcal{L}$ with messages $(\mathsf{wi}_1, \mathsf{wi}_2, \mathsf{wi}_3)$; we say that it is a WIPOK with first-message-dependent instances if it satisfies:*

1. **Completeness with first-message-dependent instances:** *For any instance choosing function X, and $\ell, n \in \mathbb{N}$,*

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

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

   The honest prover's first message wi_1 is of length n, independent of the length ℓ of the statement x.

2. **Adaptive witness-indistinguishability:** *For any polynomial $\ell(\cdot)$, non-uniform PPT verifier $V^* = \{V_n^*\}_{n\in\mathbb{N}} \in \mathbb{P}$ and all $n \in \mathbb{N}$:*

$$\Pr\left[V_n^*(x, \mathsf{wi}_1, \mathsf{wi}_2, \mathsf{wi}_3) = b \;\middle|\; \begin{array}{r} \mathsf{wi}_1 \leftarrow P(1^n, \ell(n); r) \\ x, w_0, w_1, \mathsf{wi}_2 \leftarrow V_n^*(\mathsf{wi}_1) \\ \mathsf{wi}_3 \leftarrow P(x, w_b, \mathsf{wi}_1, \mathsf{wi}_2; r) \end{array}\right] \leq \frac{1}{2} + \mathrm{negl}(n),$$

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

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

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

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

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

Instantiation. Protocols with first-message-dependent instances follow directly from the WIPOK protocol constructed in [BCPR14], assuming ZAPs and non-interactive commitments (there, the first message is taken from a fixed distribution that is completely independent of the instance).

Next, we sketch how such a protocol can be constructed without ZAPs, but assuming keyless collision-resistant hash functions, thus collapsing to an argument of knowledge against adversaries that cannot break the hash (which will anyhow be the class of interest in our zero-knowledge protocol in Sect. 3).

The Lapidot-Shamir protocol. As observed in [OV12], the Lapidot-Shamir variant of the 3-message (honest verifier) zero-knowledge protocol for Hamiltonicity [LS90a] is such that the first and second messages only depend on the size of the instance $|x| = \ell$, but not on the instance and witness themselves. The protocol, in particular, supports instances up to size ℓ that depend on the prover's first message. However, the size of the first message wi_1 in the protocol is $|\mathsf{wi}_1| > \ell$. We, on the other hand, would like to allow the instance x to be of an arbitrary polynomial size in $|\mathsf{wi}_1|$, and in particular such that $|\mathsf{wi}_1| < \ell$.

We now sketch a simple transformation from any such protocol where, in addition, the verifier's message is independent of the first prover message, into a protocol that satisfies the required first-message dependence of instances. Indeed, the verifier message in the Lapidot-Shamir protocol is simply a uniformly random string, and hence the transformation can be applied here.

The Transformation. Let $\ell(n) > n$ be any polynomial function and let $\mathcal{H}$ be a keyless collision-resistant hash function from $\{0,1\}^{\ell(n)}$ to $\{0,1\}^n$. In the new protocol $(P_{\mathsf{new}}, V_{\mathsf{new}})$, the prover computes the first message mes_1 for instances of length $\ell(n)$. Then, rather than sending mes_1 in the clear, the prover P_{new} sends $y = \mathcal{H}_n(\mathsf{mes}_1) \in \{0,1\}^n$. The verifier proceeds as in the previous protocol (P, V) (note that mes_1 is not required for it to compute mes_2). Finally the prover P_{new} answers as in the original protocol, and also sends mes_1 in the clear. The verifier V_{new} accepts, if it would in the original protocol and mes_1 is a preimage of y under $\mathcal{H}_n$.

We first note that now the size of the instance ℓ can be chosen to be an arbitrary polynomial in the length $n = |\mathsf{wi}_1|$ of the first WI message. In addition, we note that the protocol is still WI, as the view of the verifier V_{new} in the new protocol can be perfectly simulated from the view of the verifier V in the old protocol, by hashing the first message on its own.

Finally, we observe that any prover P^*_{new} that convinces the verifier in the new protocol of accepting with probability ε, can be transformed into a prover P^* that convinces the verifier of the original protocol, or to a collision-finder. Indeed, the prover P^* would first run P^*_{new} until the last message, i.e., until it obtains a valid preimage mes_1 of y. Then it would proceed interacting with V using mes_1 as its first message, and using P^*_{new} to emulate the third message. By the collision resistance of $\mathcal{H}$ the prover P^*_{new} indeed cannot make the verifier V_{new} accept with respect to two different perimages $\mathsf{mes}_1, \mathsf{mes}'_1$, except with negligible probability. Thus the prover P^* convinces V with probability $\varepsilon - \mathrm{negl}(n)$.

2.6 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. (We do not require any compactness of post-evaluation ciphertexts.)

Definition 2.5. *A scheme* $(\mathsf{Enc}, \mathsf{Eval}, \mathsf{Dec})$*, where* $\mathsf{Enc}, \mathsf{Eval}$ *are probabilistic and* Dec *is deterministic, is a semantically-secure, circuit-private, 1-hop homomorphic encryption scheme if it satisfies the following properties:*

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

$$\Pr\left[\begin{array}{c}(\mathsf{ct}, \mathsf{sk}) \leftarrow \mathsf{Enc}(x) \\ \widehat{\mathsf{ct}} \leftarrow \mathsf{Eval}(\mathsf{ct}, C) \\ \mathsf{Dec}_{\mathsf{sk}}(\widehat{\mathsf{ct}}) = C(x)\end{array}\right] = 1.$$

  where the probability is over the coin tosses of Enc *and* Eval*.*
- **Semantic security:** *For any non-uniform PPT* $\mathcal{A} = \{\mathcal{A}_n\}_{n\in\mathbb{N}} \in \mathbb{P}$*, every* $n \in \mathbb{N}$*, and any pair of inputs* $x_0, x_1 \in \{0,1\}^{\mathrm{poly}(n)}$ *of equal length,*

$$\Pr_{\substack{\mathsf{b}\leftarrow\{0,1\} \\ (\mathsf{ct},\cdot)\leftarrow\mathsf{Enc}(x_{\mathsf{b}})}}[\mathcal{A}_n(\mathsf{ct}) = \mathsf{b}] \leq \frac{1}{2} + \mathrm{negl}(n).$$

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

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

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

Instantiation. 1-hop homomorphic encryption schemes can be instantiated based on any two-message two-party computation protocol secure against semi-honest adversaries; in particular, using Yao's garbled circuits and an appropriate 2-message oblivious transfer protocol, which can be based on the Decisional Diffie-Hellman assumption, the Quadratic Residuosity assumption, or the learning with errors assumption [Yao86, GHV10, NP01, AIR01, PVW08, HK12].

3 The Protocol

In this section, we construct a 3-message ZK argument of knowledge based on 2-message memory delegation schemes. More precisely, we show that for any class of adversaries $\mathbb{A} \subseteq \mathbb{P}$, given a delegation scheme that is sound against $\mathbb{B}_1^{\mathbb{A}}$, the protocol is an argument of knowledge against $\mathbb{A}$. For simplicity we focus on classes $\mathbb{A}$ that are closed under uniform reductions; namely $\mathbb{B}_1^{\mathbb{A}} \subseteq \mathbb{A}$. These will indeed capture the adversary classes of interest for this work. We start by listing the ingredients used in the protocol, as well as introducing relevant notation.

Ingredients and notation:

– A two-message memory delegation scheme $(\mathsf{Digest}, \mathsf{Query}, \mathsf{Prov}, \mathsf{Ver})$ for γ-bounded computations, sound against provers in $\mathbb{A} \subseteq \mathbb{P}$, for a class $\mathbb{A}$ closed under uniform reductions as in Definition 2.3.
– A semantically secure and circuit-private, 1-hop homomorphic encryption scheme $(\mathsf{Enc}, \mathsf{Eval}, \mathsf{Dec})$ as in Definition 2.5.
– A 3-message WIPOK for NPwith first-message-dependent instances as in Definition 2.4. We denote its messages by $(\mathsf{wi}_1, \mathsf{wi}_2, \mathsf{wi}_3)$.
– A non-interactive perfectly-binding commitment scheme Com.
– For some $\mathsf{wi}_1, \mathsf{cmt}$, denote by $\mathcal{M}_{\mathsf{wi}_1,\mathsf{cmt}}$ a Turing machine that given memory $D = V^*$ parses V^* as a Turing machine, runs V^* on input $(\mathsf{wi}_1, \mathsf{cmt})$, parses the result as $(u, \mathsf{wi}_2, q, \widehat{\mathsf{ct}}_\tau)$, and outputs u.
– Denote by $\mathcal{V}_{\mathsf{param}}$ a circuit that has the string param hard-coded and operates as follows. Given as input a verification state τ for the delegation scheme:
  • parse $\mathsf{param} = (\mathsf{wi}_1, \mathsf{cmt}, q, u, d, t, \pi)$,
  • return 1 ("accept") if either of the following occurs:

* the delegation verifier accepts: $\mathsf{Ver}(d, \tau, \mathcal{M}_{\mathsf{wi}_1,\mathsf{cmt}}, t, u, \pi) = 1$,
* the query is inconsistent: $q \neq \mathsf{Query}(1^n; \tau)$.

In words, $\mathcal{V}_{\mathsf{param}}$, given the verification state τ, first verifies the proof π that "$\mathcal{M}_{\mathsf{wi}_1,\mathsf{cmt}}(D) = (u, \cdots)$" where D is the database corresponding to the digest d. In addition, it verifies that q is truly consistent with the coins τ. If the query is consistent, but the proof is rejected $\mathcal{V}_{\mathsf{param}}$ also rejects.

– Denote by $\mathbf{1}$ a circuit of the same size as $\mathcal{V}_{\mathsf{param}}$ that always returns 1.

We now describe the protocol in Fig. 1.

Theorem 3.1. *Given a 2-message memory delegation scheme for γ-bounded computations sound against provers in $\mathbb{A}$, a semantically-secure, circuit-private,*

Protocol 1

Common Input: an instance $x \in \mathcal{L} \cap \{0,1\}^n$, for security parameter n.
P**:** a witness $w \in \mathcal{R}_{\mathcal{L}}(x)$.

1. P computes
   – wi_1, the first message of the WIPOK for statements of length $\ell_\Psi(n)$, where ℓ_Ψ is the length of the statement Ψ defined in Step 3 below,
   – $\mathsf{cmt} \leftarrow \mathsf{Com}(0^n, 0^{\log \gamma(n)})$, a commitment to the all zero string,
   and sends $(\mathsf{wi}_1, \mathsf{cmt})$.
2. V computes
   – wi_2, the second message of the WIPOK.
   – $(\tau, q) \leftarrow \mathsf{Query}(1^n)$, verification state (w.l.o.g the coins of Query) and query,
   – $(\mathsf{ct}_\tau, \mathsf{sk}) \leftarrow \mathsf{Enc}_{\mathsf{sk}}(\tau)$, an encryption of the verification state,
   – $u \leftarrow \{0,1\}^n$, a uniformly random string,
   and sends $(u, \mathsf{wi}_2, q, \mathsf{ct}_\tau)$.
3. P computes
   – $\widehat{\mathsf{ct}} \leftarrow \mathsf{Eval}(\mathbf{1}, \mathsf{ct}_\tau)$, an evaluation of the constant one function,
   – wi_3, the third WIPOK message for the statement $\Psi = \Psi_1(x) \vee \Psi_2(\mathsf{wi}_1, \mathsf{cmt}, q, u, \mathsf{ct}_\tau, \widehat{\mathsf{ct}})$ of length $\ell_\Psi(n)$ given by:

$$\left\{ \exists w \;\middle|\; (x,w) \in \mathcal{R}_{\mathcal{L}} \right\} \bigvee$$

$$\left\{ \exists \begin{matrix} d, \pi, r_{\mathsf{cmt}} \in \{0,1\}^n \\ t \leq \gamma(n) \end{matrix} \;\middle|\; \begin{matrix} \mathsf{cmt} = \mathsf{Com}(d, t; r_{\mathsf{cmt}}) \\ \mathsf{param} = (\mathsf{wi}_1, \mathsf{cmt}, q, u, d, t, \pi) \\ \widehat{\mathsf{ct}} = \mathsf{Eval}(\mathcal{V}_{\mathsf{param}}, \mathsf{ct}_\tau) \end{matrix} \right\},$$

   using the witness $w \in \mathcal{R}_{\mathcal{L}}(x)$ for Ψ_1,
   and sends $(\widehat{\mathsf{ct}}, \mathsf{wi}_3)$.
4. V verifies the WIPOK proof $(\mathsf{wi}_1, \mathsf{wi}_2, \mathsf{wi}_3)$ for the statement Ψ and that $\mathsf{Dec}_{\mathsf{sk}}(\widehat{\mathsf{ct}}) = 1$.

Fig. 1. A 3-message ZK argument of knowledge against prover in $\mathbb{A}$.

1-hop homomorphic encryption scheme, a 3-message WIPOK with first-message-dependent instances, and a non-interactive perfectly-binding commitment scheme. The corresponding Protocol 1 (Fig. 1) is a zero-knowledge argument of knowledge against provers in $\mathbb{A}$.

Overview of proof. For simplicity, let us focus on showing that the protocol is sound and zero knowledge. (Showing it is an argument of knowledge follows a similar reasoning.) We start with soundness. Assuming that $x \notin \mathcal{L}$, in order to pass the WIPOK with respect to an evaluated cipher $\widehat{\mathsf{ct}}$ that decrypts to 1, the prover must know a digest $d \in \{0,1\}^n$, a time bound $t \leq \gamma(n)$, and proof $\pi \in \{0,1\}^n$, such that $\mathcal{V}_{\mathsf{param}}(\tau) = 1$. This, by definition, means that (d, t, π) are such that the delegation verifier Ver is convinced that the digest d corresponds to a machine V^* such that $V^*(\mathsf{wi}_1, \mathsf{cmt}) = u$. Intuitively, this implies that the prover managed to commit to a program that predicts the random string u before it was ever sent, which is unlikely. Formally, we show that such a prover can be used to break the underlying delegation scheme. Here we will also rely on the semantic security of the encryption scheme to claim that the encrypted verification state τ is hiding. Since the delegation scheme is sound against provers in $\mathbb{A}$, we shall only get soundness against such provers.

To show ZK, we construct a non-black-box simulator following the simulator of Barak [Bar01]. At high-level, the simulator uses the code of the (malicious) verifier V^* as the memory for the delegation scheme, and completes the WIPOK using *the trapdoor branch* Ψ_2 of the statement $\Psi = \Psi_1 \vee \Psi_2$. The *trapdoor witness* is basically (d, t, π), where d is the digest corresponding to V^*, $t \approx |V^*|$ and π is the corresponding delegation proof that $V^*(\mathsf{wi}_1, \mathsf{cmt}) = u$, which is now true by definition. By the perfect completeness of the delegation scheme, we know that as long as the verifier honestly encrypts some randomness τ as the private state, and gives a query q that is consistent with τ, the delegation verifier Ver will accept the corresponding proof. Thus, the circuit privacy of homomorphic evaluation (which holds also if the verifier produces a malformed ciphertext) would guarantee indistinguishability from a real proof, where the prover actually evaluates the constant **1** circuit.

A detailed proof follows. We first prove in Sect. 3.1 that the protocol is an argument of knowledge. Then we prove in Sect. 3.2 that the protocol is zero knowledge.

3.1 Proving that the Protocol Is an Argument of Knowledge

In this section, we show that the protocol is an argument of knowledge against provers in $\mathbb{A}$.

Proposition 3.1. *Protocol 1 (Fig. 1) is an argument of knowledge against provers in* $\mathbb{A}$.

Proof. We show that there exists a uniform PPT extractor $\mathcal{E} \in \mathbb{B}_1$ and a uniform PPT reduction $\mathcal{R} \in \mathbb{B}_1$, such that for any prover $P^* = \{P^*_n\}_{n\in\mathbb{N}} \in \mathbb{A}$ that

generates $x_n \in \{0,1\}^n$ and convinces V of accepting x_n with non-negligible probability $\varepsilon(n)$, one of the following holds:

– $\mathcal{E}^{P_n^*}(1^{1/\varepsilon(n)}, x_n)$ outputs $w \in \mathcal{R}_{\mathcal{L}}(x_n)$ with probability $\varepsilon(n)^2/4 - \mathrm{negl}(n)$,[6] or
– $\mathcal{R}^{P_n^*}$ breaks the soundness of the delegation scheme with probability $n^{-O(1)}$.

We start by describing the extractor. Throughout the description (and following proof), we will often omit n, when it is clear from the context.

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

1. Derives from P^* a new prover P^*_{wi} for the WIPOK as follows. P^*_{wi} emulates the role of P^* in the WIPOK; in particular, it would (honestly) sample $(\tau, (\mathsf{sk}, \mathsf{ct}_\tau), u)$ on its own to compute the second verifier message $(\mathsf{wi}_2, q, \mathsf{ct}_\tau, u)$ that P^* receives.
2. Chooses the random coins r for the new prover P^*_{wi}, and samples a transcript $\mathsf{tr} = (\Psi, \mathsf{wi}_1, \mathsf{wi}_2, \mathsf{wi}_3)$ of an execution with the honest WIPOK verifier V_{wi}.
3. Applies the WIPOK extractor $\mathcal{E}_{\mathsf{wi}}$ on the transcript tr, with oracle access to P^*_{wi}, and extraction parameter $2/\varepsilon$. That is, computes $w \leftarrow \mathcal{E}_{\mathsf{wi}}^{P^*_{\mathsf{wi}}(r)}(1^{2/\varepsilon}, \mathsf{tr})$.
4. Outputs w.

Our strategy will be to show the required reduction $\mathcal{R}$, such that if the extractor fails to extract with the required probability, then the reduction breaks the underlying delegation scheme. Thus from hereon, we assume that for some noticeable function $\eta(n) = n^{-O(1)}$, with probability at most $\varepsilon^2/4 - \eta$ the extracted witness w is in $\mathcal{R}_{\mathcal{L}}(x)$. Rather than already describing the reduction $\mathcal{R}$, we shall first establish several claims regarding the extraction procedure and the consequences of extraction failure. These will motivate our concrete construction of the reduction $\mathcal{R}$.

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

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

Recall that every execution of $\mathcal{E}_{\mathsf{wi}}$ induces a choice r for P^*_{wi}, a WIPOK transcript $\mathsf{tr} = (\Psi, \mathsf{wi}_1, \mathsf{wi}_2, \mathsf{wi}_3)$, and values $(\mathsf{cmt}, q, u, \mathsf{ct}_\tau, \widehat{\mathsf{ct}})$ exchanged in the induced interaction between the zero-knowledge prover P^* and the zero-knowledge verifier V. These values, in turn, determine the formula

$$\Psi = \Psi_1(x) \vee \Psi_2(\mathsf{wi}_1, \mathsf{cmt}, q, u, \mathsf{ct}_\tau, \widehat{\mathsf{ct}}).$$

[6] We note that the extraction probability can then be amplified to $1 - \mathrm{negl}(n)$ by standard repetition.

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

Claim 3.2 (Extraction for good r). *For any* good r *for* P^*_{wi}*, it holds that* w *satisfies the induced statement* Ψ *and* $\mathsf{Dec}_{\mathsf{sk}}(\widehat{\mathsf{ct}}) = 1$ *with probability* $\varepsilon(n)/2 - \mathrm{negl}(n)$ *over a transcript* tr*, and coins for* $\mathcal{E}_{\mathsf{wi}}$*.*

Proof of Claim 3.2. Fix some good coins r. Since the coins r are good, the WIPOK verifier V_{wi} is convinced by P^*_{wi} with probability at least $\varepsilon/2$, meaning that V_{wi} accepts and in addition $\mathsf{Dec}_{\mathsf{sk}}(\widehat{\mathsf{ct}}) = 1$. We claim that when this occurs then, except with probability $\mathrm{negl}(n)$, the extractor $\mathcal{E}_{\mathsf{wi}}$, also outputs a valid witness w for Ψ. This follows directly from the extraction guarantee of the WIPOK. □

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

Claim 3.3 (Extracting a trapdoor witness). *In a random execution of the extractor, the extracted witness* w *satisfies the trapdoor statement, namely* $\Psi_2(\mathsf{wi}_1, \mathsf{cmt}, q, u, \mathsf{ct}_\tau, \widehat{\mathsf{ct}})$*, and in addition* $\mathsf{Dec}_{\mathsf{sk}}(\widehat{\mathsf{ct}}) = 1$*, with probability at least* $\eta(n) - \mathrm{negl}(n)$ *over the choice of* r *for* P^*_{wi}*, a transcript* tr*, and coins for* $\mathcal{E}_{\mathsf{wi}}$*.*

Proof of Claim 3.3. First, by the $(\varepsilon/2)$-density of good r's and Claim 3.2, we deduce that in a random execution the extracted w satisfies the statement $\Psi = \Psi_1 \vee \Psi_2$, and in addition $\mathsf{Dec}_{\mathsf{sk}}(\widehat{\mathsf{ct}}) = 1$, with probability at least $\varepsilon^2/4 - \mathrm{negl}(n)$. Combining this with the fact that $w \in \mathcal{R}_{\mathcal{L}}(x)$ with probability at most $\varepsilon^2/4 - \eta$, the claim follows. □

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

$$w = (d, \pi, t, r_{\mathsf{cmt}}) : \begin{array}{l} d, \pi \in \{0,1\}^n, t \leq \gamma(n) \\ \widehat{\mathsf{ct}} = \mathsf{Eval}(\mathcal{V}_{\mathsf{param}}, \mathsf{ct}_\tau) \\ \mathsf{param} = (\mathsf{wi}_1, \mathsf{cmt}, q, u, d, t, \pi) \\ \mathsf{cmt} = \mathsf{Com}(d, t; r_{\mathsf{cmt}}) \end{array}.$$

Furthermore, by the definition of $\mathcal{V}_{\mathsf{param}}$ and the perfect completeness of the 1-hop homomorphic encryption,

$$\mathsf{Dec}_{\mathsf{sk}}(\widehat{\mathsf{ct}}) = \mathcal{V}_{\mathsf{param}}(\tau) = \mathsf{Ver}(d, \tau, \mathcal{M}_{\mathsf{wi}_1, \mathsf{cmt}}, t, u, \pi).$$

We can thus deduce that, with probability η, the witness $w = (d, \pi, t, r_{\mathsf{cmt}})$ extracted by $\mathcal{E}$ is such that: (a) $\mathsf{Ver}(d, \tau, \mathcal{M}_{\mathsf{wi}_1, \mathsf{cmt}}, t, u, \pi) = 1$, and (b) $\mathsf{cmt} = \mathsf{Com}(d, t; r_{\mathsf{cmt}})$.

An equivalent experiment that hides the secret verification state τ. We now consider an augmented extraction procedure $\mathcal{E}_{\mathsf{aug}} \in \mathbb{B}_1$ that behaves

exactly as the original extractor $\mathcal{E}$, except that, when P^*_{wi} emulates P^*, it does not sample an encryption ct_τ of the secret verification state τ, but rather it samples an encryption ct_0 of $0^{|\tau|}$. We claim that in this alternative experiment, the above two conditions (a) and (b) still hold with the same probability up to a negligible difference.

Claim 3.4 (Convincing probability in alternative experiment). *With probability* $\eta - \mathrm{negl}(n)$*, the witness* $w = (d, \pi, t, r_{\mathsf{cmt}})$ *extracted by* $\mathcal{E}_{\mathsf{aug}}$ *is such that:*
(a) $\mathsf{Ver}(d, \tau, \mathcal{M}_{\mathsf{wi}_1,\mathsf{cmt}}, t, u, \pi) = 1$*, and (b)* $\mathsf{cmt} = \mathsf{Com}(d, t; r_{\mathsf{cmt}})$.

Proof sketch of Claim 3.4. This claim follows from the semantic security of the 1-hop homomorphic encryption scheme. Indeed, if the above was not the case, we can distinguish between an encryption of τ and one of $0^{|\tau|}$. For this, note that the first experiment with ct_τ (respectively, the second with ct_0) can be perfectly emulated given τ and the ciphertext ct_τ (respectively, ct_0), and in addition the above two conditions (a) and (b) can be tested efficiently. □

The reduction $\mathcal{R}$ to the soundness of delegation. We are now ready to describe the reduction $\mathcal{R}$ that breaks the soundness of the delegation scheme. In what follows, we view the randomness r for P^*_{wi} as split into $r = (r_1, \tau, u, r_2)$, where r_1 is any randomness used to generate the first prover message $(\mathsf{wi}_1, \mathsf{cmt})$, τ is the randomness for Query and u is the random string both used to emulate the second verifier message, and r_2 are any additional random coins used by P^*_{wi}. The reduction $\mathcal{R}^{P^*_n}(1^{1/\varepsilon(n)}, x_n)$ breaks the delegation scheme as follows:[7]

1. Samples $r^* = (r^*_1, \tau^*, u^*, r^*_2)$ uniformly at random.
2. Runs $\mathcal{E}^{P^*}_{\mathsf{aug}}(1^{1/\varepsilon}, x)$ using r^* as the randomness for P_{wi}. Let $(\mathsf{cmt}^*, \mathsf{wi}^*_1)$ be the corresponding first prover message (which is completely determined by the choice of r^*_1), and let $w^* = (d^*, \pi^*, t^*, r^*_{\mathsf{cmt}})$ be the witness output by the extractor.
3. Samples $u, u' \leftarrow \{0,1\}^n$ uniformly at random.
4. Declares d^* as the digest, $\mathcal{M}_{\mathsf{wi}^*_1,\mathsf{cmt}^*}$ as the machine to be evaluated over the memory, t^* the bound on its running time, and (u, u') as the two outputs for the attack.
5. Given a delegation query q, $\mathcal{R}$ generates two proofs π and π' for u and u' respectively as follows:
   (a) Samples $r = (r^*_1, \perp, u, r_2)$ and $r' = (r^*_1, \perp, u', r'_2)$, where in both r^*_1 is the same randomness sampled before, (u, u') are the random strings sampled before, and (r_2, r'_2) are uniformly random strings.
   (b) Runs $\mathcal{E}^{P^*}_{\mathsf{aug}}(1^{1/\varepsilon}, x)$ once with respect to r and another time with respect to r', with one exception—the prover P^*_{wi} constructed by $\mathcal{E}^{P^*}_{\mathsf{aug}}$ does not emulate on its own the delegation query in the verifier's message, but

[7] Here we give the reduction $(1^{1/\varepsilon(n)}, x_n)$ for the sake of simplicity and clarity of exposition. Recall that x_n is generated by P^*_n. Also, ε can be approximated by sampling. Thus the reduction can (uniformly) obtain these two inputs from P^*.

rather it uses the external query q that $\mathcal{R}$ is given. The two executions of $\mathcal{E}_{\mathsf{aug}}^{P^*}$ then produce witnesses $w = (d, \pi, t, r_{\mathsf{cmt}})$ and $w' = (d', \pi', t', r'_{\mathsf{cmt}})$.

(c) Output (π, π').

We first note that the running time of $\mathcal{R}$ is polynomial in n and in the running of $\mathcal{E}_{\mathsf{aug}}$, which is in turn polynomial in the running time of P^* and in $1/\varepsilon(n) = n^{O(1)}$. Thus it is overall polynomial in n.

To complete the proof, we show that $\mathcal{R}$ breaks the scheme with noticeable probability.

Claim 3.5. *$u \neq u'$ and π and π' both convince the delegation verifier with probability $\Omega(\eta(n)^5)$.*

Proof of Claim 3.5. Throughout, let us denote by G the event that the witness $w = (d, \pi, t, r_{\mathsf{cmt}})$ extracted by $\mathcal{E}_{\mathsf{aug}}$ is such that: (a) $\mathsf{Ver}(d, \tau, \mathcal{M}_{\mathsf{wi}_1,\mathsf{cmt}}, t, u, \pi) = 1$, and (b) $\mathsf{cmt} = \mathsf{Com}(d, t; r_{\mathsf{cmt}})$. We will call r_1^* good_1, if with probability $\eta/2$ (over all other randomness), G occurs. Then by Claim 3.4 and averaging, with probability $\eta/2 - \mathrm{negl}(n)$ over a choice of a random r_1^*, it is good_1. Next, for a fixed r_1^* and τ, we will say that τ is r_1^*-good, if with probability $\eta/4$ over a choice of random (u, r_2^*), $G0$ occurs. Then, by averaging, for any good_1 r_1^*, with probability $\eta/4 - \mathrm{negl}(n)$ over a choice of a random τ, it is r_1^*-good.

We are now ready to lower bound the probability that $\mathcal{R}$ breaks the delegation scheme. This is based on the following assertions:

1. In Step 1, with probability $\eta/2 - \mathrm{negl}(n)$, $\mathcal{R}$ samples a good_1 r_1^*.
2. Conditioned on r_1^* being good_1:
   (a) In Step 2, with probability $\eta/2$, G occurs. In particular, the extracted $(d^*, t^*, r_{\mathsf{cmt}}^*)$ are valid in the sense that $\mathsf{cmt}^* = \mathsf{Com}(d^*, t^*; r_{\mathsf{cmt}}^*)$, cmt^* is the commitment generated in the first prover message (determined by the choice of r_1^*).
   (b) In Step 5, with probability $\eta/4 - \mathrm{negl}(n)$, the coins τ chosen by the delegation Query algorithm (inducing the query q) are r_1^*-good.
   (c) Conditioned on the coins τ of Query being r_1^*-good:
      i. In Step 5, with probability $\eta/4$, the event G occurs. Thus the extracted $(d, t, r_{\mathsf{cmt}}, \pi)$ are valid in the sense that $\mathsf{cmt}^* = \mathsf{Com}(d, t; r_{\mathsf{cmt}})$, as well as $\mathsf{Ver}(d, \tau, \mathcal{M}_{\mathsf{wi}_1^*,\mathsf{cmt}^*}, t, u, \pi) = 1$. Recall that $(\mathsf{wi}_1^*, \mathsf{cmt}^*)$ are generated in the first prover message (and are determined by the choice of r_1^*).
      ii. The same holds independently for the second random output u'.
3. In Step 3, with probability $1 - 2^{-n}$, the outputs u, u' sampled by $\mathcal{R}$ are distinct.
4. If $\mathsf{cmt}^* = \mathsf{Com}(d^*, t^*; r_{\mathsf{cmt}}^*) = \mathsf{Com}(d, t; r_{\mathsf{cmt}}) = \mathsf{Com}(d', t'; r'_{\mathsf{cmt}})$, then $(d, t) = (d', t') = (d^*, t^*)$.

The first two assertions follow directly from the definitions and averaging arguments made above. The third assertion follows from the collision probability of

two random strings of length n. The last assertion follows from the fact that the commitment Com is perfectly binding.

It is left to note that if all of the above occur, then $\mathcal{R}$ manages to produce accepting proofs (π, π') for two different outcomes (u, u') with respect to the same digest d^* and machine $\mathcal{M}_{\mathsf{wi}_1^*, \mathsf{cmt}^*}$; thus, it breaks soundness. This happens with probability

$$\left(\frac{\eta}{2} - \mathrm{negl}(n)\right) \cdot \frac{\eta}{2} \cdot \left(\frac{\eta}{4} - \mathrm{negl}(n)\right) \cdot \left(\frac{\eta}{4}\right)^2 - 2^{-n} = \Omega(\eta^5).$$

This completes the proof of Claim 3.5. □
This completes the proof of Proposition 3.1.

3.2 Proving that the Protocol Is Zero Knowledge

In this section, we prove

Proposition 3.2. *Protocol 1 (Fig. 1) is* ZK *against non-uniform PPT verifiers.*

Proof. We describe a universal ZK simulator $\mathcal{S}$ that given the code of any non-uniform PPT $V^* = \{V_n^*\}_{n\in\mathbb{N}}$, a polynomial bound $t(n) = n^{O(1)}$ on its running time (or more precisely the time required for a universal machine to run it), and $x \in \mathcal{L}$, simulates the view of V. We shall assume V^* is deterministic; this is w.l.o.g as we can always sample random coins for V^* and hardwire them into its non-uniform description. Throughout, we often omit the security parameter n when clear from the context.

The simulator $\mathcal{S}(V_n^*, t(n), x)$, where $|x| = n$, operates as follows:

1. Generates the first message $(\mathsf{wi}_1, \mathsf{cmt})$ as follows:
   (a) Samples a first message $\mathsf{wi}_1 \in \{0,1\}^n$ of the WIPOK.
   (b) Computes a digest $d = \mathsf{Digest}(1^n, V^*)$ of the verifier's code.
   (c) Computes a commitment $\mathsf{cmt} = \mathsf{Com}(d, t; r_{\mathsf{cmt}})$ to the digest d and V^*'s running time t, using random coins $r_{\mathsf{cmt}} \leftarrow \{0,1\}^n$. Here t is interpreted as string in $\{0,1\}^{\log \gamma(n)}$. This is possible, for all large enough n, as $t(n) = n^{O(1)} \ll n^{\omega(1)} = \gamma(n)$.
2. Runs the verifier to obtain $(\mathsf{wi}_2, q, u, \mathsf{ct}_\tau) \leftarrow V^*(\mathsf{wi}_1, \mathsf{cmt})$.
3. Computes the third message $(\widehat{\mathsf{ct}}, \mathsf{wi}_3)$ as follows:
   (a) Computes a proof $\pi = \mathsf{Prov}(1^t, \mathcal{M}_{\mathsf{wi}_1, \mathsf{cmt}}, V^*, q)$ that the digested code of V^* outputs u.
   (b) Samples $\widehat{\mathsf{ct}} \leftarrow \mathsf{Eval}(\mathcal{V}_{\mathsf{param}}, \mathsf{ct}_\tau)$, for $\mathsf{param} = (\mathsf{wi}_1, \mathsf{cmt}, q, u, d, t, \pi)$.
   (c) Computes the third WIPOK message wi_3 for the statement $\Psi = \Psi_1(x) \vee \Psi_2(\mathsf{wi}_1, \mathsf{cmt}, q, u, \mathsf{ct}_\tau, \widehat{\mathsf{ct}})$ given by:

$$\left\{ \exists w \;\middle|\; \begin{array}{l}(x,w)\\ \in \mathcal{R}_{\mathcal{L}}\end{array} \right\} \bigvee \left\{ \exists \begin{array}{l} d, \pi, r_{\mathsf{cmt}} \in \{0,1\}^n \\ t \le \gamma(n) \end{array} \;\middle|\; \begin{array}{l} \widehat{\mathsf{ct}} = \mathsf{Eval}(\mathcal{V}_{\mathsf{param}}, \mathsf{ct}_\tau) \\ \mathsf{param} = (\mathsf{wi}_1, \mathsf{cmt}, q, u, d, t, \pi) \\ \mathsf{cmt} = \mathsf{Com}(d, t; r_{\mathsf{cmt}}) \end{array} \right\},$$

   using the witness $(d, \pi, r_{\mathsf{cmt}}, t)$ for the trapdoor statement Ψ_2.

(d) Outputs the view $(\mathsf{wi}_1, \mathsf{cmt}, \widehat{\mathsf{ct}}, \mathsf{wi}_3)$ of V^*.

We now show that the view generated by $\mathcal{S}$ is computationally indistinguishable from the view of V^* in an execution with the honest prover P. We do this by exhibiting a sequence of hybrids.

Hybrid 1: The view $(\mathsf{wi}_1, \mathsf{cmt}, \widehat{\mathsf{ct}}, \mathsf{wi}_3)$ is generated by $\mathcal{S}$.

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

Hybrid 3: Instead of generating cmt as a commitment $\mathsf{cmt} = \mathsf{Com}(d, t; r_{\mathsf{cmt}})$ to (d, t), it is generated as a commitment to $0^{n+\log \gamma(n)}$. Note that in this hybrid the commitment's randomness r_{cmt} is not used anywhere, but in the generation of cmt. Thus, by the computational hiding of the commitment, this hybrid is computationally indistinguishable from Hybrid 2.

Hybrid 4: The view $(\mathsf{wi}_1, \mathsf{cmt}, \widehat{\mathsf{ct}}, \mathsf{wi}_3)$ is generated in an interaction of V^* with the honest prover P. The difference from Hybrid 3 is in that $\widehat{\mathsf{ct}}$ is sampled from $\mathsf{Eval}(\mathbf{1}, \mathsf{ct}_\tau)$ instead of $\mathsf{Eval}(\mathcal{V}_{\mathsf{param}}, \mathsf{ct}_\tau)$. First, note that by the perfect completeness of the delegation scheme, for any $\tau \in \{0,1\}^n$, $\mathcal{V}_{\mathsf{param}}(\tau) = \mathbf{1}(\tau) = 1$. Indeed, by definition we know that

$$\mathcal{M}_{\mathsf{wi}_1, \mathsf{cmt}}(V^*) = V^*(\mathsf{wi}_1, \mathsf{cmt})[1] = u,$$

and this output is produced after at most t steps. Thus, assuming that $q = \mathsf{Query}(1^n; \tau)$, the delegation verifier accepts; namely, $\mathsf{Ver}(d, \tau, \mathcal{M}_{\mathsf{wi}_1, \mathsf{cmt}}, t, u, \pi) = 1$, and by definition $\mathcal{V}_{\mathsf{param}}(\tau) = 1$. Also, if $q \neq \mathsf{Query}(1^n; \tau)$, then $\mathcal{V}_{\mathsf{param}}(\tau) = 1$ by definition.

By the circuit privacy of the 1-hop homomorphic encryption, the above guarantees indistinguishability whenever ct_τ is a well-formed ciphertext since

$$\begin{aligned}\mathsf{Eval}(\mathcal{V}_{\mathsf{param}}, \mathsf{ct}_\tau) \approx_c \mathcal{S}_{\mathsf{1hop}}(\mathsf{ct}_\tau, \mathcal{V}_{\mathsf{param}}(\tau), |\mathcal{V}_{\mathsf{param}}|) \equiv \\ \mathcal{S}_{\mathsf{1hop}}(\mathsf{ct}_\tau, \mathbf{1}(\tau), |\mathbf{1}|) \approx_c \mathsf{Eval}(\mathbf{1}, \mathsf{ct}_\tau).\end{aligned}$$

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

$$\mathsf{Eval}(\mathcal{V}_{\mathsf{param}}, \mathsf{ct}^*) \approx_c \mathcal{S}_{\mathsf{1hop}}(\mathsf{ct}^*, \bot, |\mathcal{V}_{\mathsf{param}}|) \equiv \mathcal{S}_{\mathsf{1hop}}(\mathsf{ct}^*, \bot, |\mathbf{1}|) \approx_c \mathsf{Eval}(\mathbf{1}, \mathsf{ct}^*).$$

It follows that Hybrid 4 is computationally indistinguishable from Hybrid 3.

This completes the proof of Proposition 3.2.

Acknowledgments. We thank Ran Canetti, Shai Halevi and Hugo Krawczyk for helpful comments and for pointing out the connection to [Rog06].

References

[AIR01] Aiello, B., Ishai, Y., Reingold, O.: Priced oblivious transfer: how to sell digital goods. In: Pfitzmann, B. (ed.) EUROCRYPT 2001. LNCS, vol. 2045, pp. 119–135. Springer, Heidelberg (2001). doi:10.1007/3-540-44987-6_8

[Bar01] Barak, B.: How to go beyond the black-box simulation barrier. In: FOCS, pp. 106–115 (2001)

[BCC88] Brassard, G., Chaum, D., Crépeau, C.: Minimum disclosure proofs of knowledge. J. Comput. Syst. Sci. **37**(2), 156–189 (1988)

[BCC+14] Bitansky, N., Canetti, R., Chiesa, A., Goldwasser, S., Lin, H., Rubinstein, A., Tromer, E.: The hunting of the SNARK. IACR Cryptology ePrint Archive, 2014:580 (2014)

[BCPR13] Bitansky, N., Canetti, R., Paneth, O., Rosen, A.: More on the impossibility of virtual-black-box obfuscation with auxiliary input. IACR Cryptology ePrint Archive, 2013:701 (2013)

[BCPR14] Bitansky, N., Canetti, R., Paneth, O., Rosen, A.: On the existence of extractable one-way functions. In: Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31–June 03 2014, pp. 505–514 (2014)

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

[BGGL01] Barak, B., Goldreich, O., Goldwasser, S., Lindell, Y.: Resettably-sound zero-knowledge and its applications. In: FOCS, pp. 116–125 (2001)

[BHK16] Brakerski, Z., Holmgren, J., Kalai, Y., Non-interactive ram, batch np delegation from any pir. Cryptology ePrint Archive, Report 2016/459 (2016). http://eprint.iacr.org/

[BJY97] Bellare, M., Jakobsson, M., Yung, M.: Round-optimal zero-knowledge arguments based on any one-way function. In: Fumy, W. (ed.) EUROCRYPT 1997. LNCS, vol. 1233, pp. 280–305. Springer, Heidelberg (1997). doi:10.1007/3-540-69053-0_20

[BM14] Brzuska, C., Mittelbach, A.: Indistinguishability obfuscation versus multi-bit point obfuscation with auxiliary input. In: Sarkar, P., Iwata, T. (eds.) ASIACRYPT 2014. LNCS, vol. 8874, pp. 142–161. Springer, Heidelberg (2014). doi:10.1007/978-3-662-45608-8_8

[BP04a] Barak, B., Pass, R.: On the possibility of one-message weak zero-knowledge. In: Naor, M. (ed.) TCC 2004. LNCS, vol. 2951, pp. 121–132. Springer, Heidelberg (2004). doi:10.1007/978-3-540-24638-1_7

[BP04b] Bellare, M., Palacio, A.: The knowledge-of-exponent assumptions and 3-round zero-knowledge protocols. In: Franklin, M. (ed.) CRYPTO 2004. LNCS, vol. 3152, pp. 273–289. Springer, Heidelberg (2004). doi:10.1007/978-3-540-28628-8_17

[BP12] Bitansky, N., Paneth, O.: From the impossibility of obfuscation to a new non-black-box simulation technique. In: FOCS (2012)

[BP13] Bitansky, N., Paneth, O.: On the impossibility of approximate obfuscation and applications to resettable cryptography. In: STOC, pp. 241–250 (2013)

[CD09] Canetti, R., Dakdouk, R.R.: Towards a theory of extractable functions. In: Reingold, O. (ed.) TCC 2009. LNCS, vol. 5444, pp. 595–613. Springer, Heidelberg (2009). doi:10.1007/978-3-642-00457-5_35

[CDPW07] Canetti, R., Dodis, Y., Pass, R., Walfish, S.: Universally composable security with global setup. In: Vadhan, S.P. (ed.) TCC 2007. LNCS, vol. 4392, pp. 61–85. Springer, Heidelberg (2007). doi:10.1007/978-3-540-70936-7_4

[CKLR11] Chung, K.-M., Kalai, Y.T., Liu, F.-H., Raz, R.: Memory delegation. In: Rogaway, P. (ed.) CRYPTO 2011. LNCS, vol. 6841, pp. 151–168. Springer, Heidelberg (2011). doi:10.1007/978-3-642-22792-9_9

[CLP13a] Canetti, R., Lin, H., Paneth, O.: Public-coin concurrent zero-knowledge in the global hash model. In: Sahai, A. (ed.) TCC 2013. LNCS, vol. 7785, pp. 80–99. Springer, Heidelberg (2013). doi:10.1007/978-3-642-36594-2_5

[CLP13b] Chung, K.-M., Lin, H., Pass, R.: Constant-round concurrent zero knowledge from p-certificates. In: FOCS (2013)

[CLP15] Chung, K.-M., Lin, H., Pass, R.: Constant-round concurrent zero-knowledge from indistinguishability obfuscation. In: Gennaro, R., Robshaw, M. (eds.) CRYPTO 2015. LNCS, vol. 9215, pp. 287–307. Springer, Heidelberg (2015). doi:10.1007/978-3-662-47989-6_14

[COP+14] Chung, K.-M., Ostrovsky, R., Pass, R., Venkitasubramaniam, M., Visconti, I.: 4-round resettably-sound zero knowledge. In: Lindell, Y. (ed.) TCC 2014. LNCS, vol. 8349, pp. 192–216. Springer, Heidelberg (2014). doi:10.1007/978-3-642-54242-8_9

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

[FS89] Feige, U., Shamir, A.: Zero knowledge proofs of knowledge in two rounds. In: Brassard, G. (ed.) CRYPTO 1989. LNCS, vol. 435, pp. 526–544. Springer, Heidelberg (1990). doi:10.1007/0-387-34805-0_46

[GHV10] Gentry, C., Halevi, S., Vaikuntanathan, V.: i-Hop homomorphic encryption and rerandomizable yao circuits. In: CRYPTO, pp. 155–172 (2010)

[GK96a] Goldreich, O., Kahan, A.: How to construct constant-round zero-knowledge proof systems for NP. J. Cryptol. **9**(3), 167–190 (1996)

[GK96b] Goldreich, O., Krawczyk, H.: On the composition of zero-knowledge proof systems. SIAM J. Comput. **25**(1), 169–192 (1996)

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

[GMW91] Goldreich, O., Micali, S., Wigderson, A.: Proofs that yield nothing but their validity for all languages in NP have zero-knowledge proof systems. J. ACM **38**(3), 691–729 (1991)

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

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

[HK12] Halevi, S., Kalai, Y.T.: Smooth projective hashing and two-message oblivious transfer. J. Cryptol. **25**(1), 158–193 (2012)

[HT98] Hada, S., Tanaka, T.: On the existence of 3-round zero-knowledge protocols. In: Krawczyk, H. (ed.) CRYPTO 1998. LNCS, vol. 1462, pp. 408–423. Springer, Heidelberg (1998). doi:10.1007/BFb0055744

[HV16] Hazay, C., Venkitasubramaniam, M.: On the power of secure two-party computation. Cryptology ePrint Archive, Report 2016/074 (2016). http://eprint.iacr.org/

[IKOS09] Ishai, Y., Kushilevitz, E., Ostrovsky, R., Sahai, A.: Zero-knowledge proofs from secure multiparty computation. SIAM J. Comput. **39**(3), 1121–1152 (2009)

[Kat12] Katz, J.: Which languages have 4-round zero-knowledge proofs? J. Cryptol. **25**(1), 41–56 (2012)

[KP15] Kalai, Y.T., Paneth, O.: Delegating ram computations. Cryptology ePrint Archive, Report 2015/957 (2015). http://eprint.iacr.org/

[KRR14] Kalai, Y.T., Raz, R., Rothblum, R.D.: How to delegate computations: the power of no-signaling proofs. In: Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31–June 03 2014, pp. 485–494 (2014)

[LS90a] Lapidot, D., Shamir, A.: Publicly verifiable non-interactive zero-knowledge proofs. In: Menezes, A.J., Vanstone, S.A. (eds.) CRYPTO 1990. LNCS, vol. 537, pp. 353–365. Springer, Heidelberg (1991). doi:10.1007/3-540-38424-3_26

[NP01] Naor, M., Pinkas, B.: Efficient oblivious transfer protocols. In: SODA, pp. 448–457 (2001)

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

[Pas03] Pass, R.: Simulation in quasi-polynomial time, and its application to protocol composition. In: Biham, E. (ed.) EUROCRYPT 2003. LNCS, vol. 2656, pp. 160–176. Springer, Heidelberg (2003). doi:10.1007/3-540-39200-9_10

[PVW08] Peikert, C., Vaikuntanathan, V., Waters, B.: A framework for efficient and composable oblivious transfer. In: Wagner, D. (ed.) CRYPTO 2008. LNCS, vol. 5157, pp. 554–571. Springer, Heidelberg (2008). doi:10.1007/978-3-540-85174-5_31

[Reg09] Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. J. ACM **56**(6), 34.1–34.40 (2009)

[Rog06] Rogaway, P.: Formalizing human ignorance. In: Nguyen, P.Q. (ed.) VIETCRYPT 2006. LNCS, vol. 4341, pp. 211–228. Springer, Heidelberg (2006). doi:10.1007/11958239_14

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