Weak Zero-Knowledge Beyond the Black-Box Barrier

Nir Bitansky[*] Omer Paneth[†]

September 23, 2018

Abstract

The round complexity of zero-knowledge protocols is a long-standing open question, yet to be settled under standard assumptions. So far, the question has appeared equally challenging for relaxations such as weak zero-knowledge and witness hiding. Protocols satisfying these relaxed notions, under standard assumptions, have at least four messages, just like full-fledged zero knowledge. The difficulty in improving round complexity stems from a fundamental barrier: non of these notions can be achieved in three messages via reductions (or simulators) that treat the verifier as a black box.

We introduce a new non-black-box technique and use it to obtain the first protocols that cross this barrier under standard assumptions. Our main results are:

- Weak zero-knowledge for **NP** in three messages, assuming fully-homomorphic encryption, and other standard primitives (known for example under the Learning with Errors assumption).
- Weak zero-knowledge for **NP** ∩ **coNP** in two messages, assuming in addition non-interactive witness-indistinguishable proofs.

We also give witness-hiding protocol for **NP** in two-message, assuming in addition witness encryption. This protocol is also publicly verifiable.

Our technique is based on a new *homomorphic trapdoor paradigm*, which can be seen as a non-black-box analog of the classical Feige-Lapidot-Shamir trapdoor paradigm.

[*] Tel Aviv University, nirbitan@tau.ac.il. Member of the Check Point Institute of Information Security. Supported by the Alon Young Faculty Fellowship and by Len Blavatnik and the Blavatnik Family foundation.

[†] MIT, omerpa@mit.edu Supported by NSF Grants CNS-1350619 and CNS-1414119, and the Defense Advanced Research Projects Agency (DARPA) and the U.S. Army Research Office under contracts W911NF-15-C-0226 and W911NF-15-C-0236.Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the DARPA and ARO. Part of this research was done while visiting Tel Aviv university and supported by Len Blavatnik and the Blavatnik Family foundation.

Contents

1 Introduction

Zero-knowledge protocols are spectacular. They allow to prove any **NP** statement without revealing anything but the statement's validity. That is, whatever a malicious verifier learns from the protocol can be efficiently simulated from the statement alone, without ever interacting with the prover. Since their invention [GMR89] and construction for all of **NP** [GMW91], zero-knowledge protocols have become an essential building block in cryptographic protocols, and have had a profound impact on modern cryptography.

A central question in the study of zero knowledge is that of *round complexity*. Zero-knowledge arguments with a negligible soundness error can be achieved in four messages [FS90], under the minimal assumption of one-way functions [BJY97].[1] In terms of lower bounds, zero-knowledge arguments for languages outside **BPP**, and without any trusted setup, require at least three messages [GO94].

Zero knowledge protocols with an optimal number of messages (and a negligible soundness error) have been pursued over the last three decades and have proven difficult to construct. Three-message zero-knowledge arguments were only constructed under *auxiliary-input knowledge assumptions*, which are considered implausible [HT98, BP04b, BCPR14, BP15c, BCC$^+$17], and more recently, under a new, non-standard, assumption on the multi-collision resistance of keyless hash functions [BKP18]. Three-message protocols based on standard cryptographic assumptions remain out of reach.

Relaxing zero knowledge. Given the current state of affairs, it is natural to consider relaxations of the zero-knowledge privacy guarantee. Three main relaxations considered in the literature are:

- **Witness indistinguishability [FS90]:** Ensures that a malicious verifier cannot distinguish between proofs that are generated using different witnesses. While natural and often a useful building block in applications, witness indistinguishability is still quite limited. For example, in the rather common scenario where statements have a unique witness, witness indistinguishability becomes meaningless.

- **Witness hiding [FS90]:** Ensures that a malicious verifier cannot learn an entire witness from the proof; that is, unless such a witness can be efficiently computed from the statement alone. In contrast to witness indistinguishability, the witness hiding requirement is also meaningful in the unique witness case.

- **Weak zero-knowledge [DNRS03]:** Relaxes zero-knowledge by switching quantifiers. Full-fledged zero-knowledge requires that for every verifier there exists a simulator that generates an interaction, indistinguishable from a real one, for every distinguisher. In contrast, weak zero-knowledge requires that for every verifier and distinguisher, there exists a simulator that fools this specific distinguisher. We also allow the simulator to depend on the desired distinguishing gap.[2]

  Weak zero-knowledge hides any joint predicate of the statement and witness (used by the prover) that cannot be computed from the statement alone. It implies both witness hiding and witness indistinguishability.

The above relaxations are not subject to the same lower bounds as full-fledged zero-knowledge. In fact, the only known unconditional lower bound rules out weak zero-knowledge in one message [GO94].

As for constructions, witness indistinguishability has indeed been obtained, under standard assumptions, in three [GMW91, FS90], two [DN07, BGI$^+$17, JKKR17], and eventually even one message [BOV07, GOS12]. In contrast, weak zero-knowledge and witness hiding have proven to be just as challenging to construct as full-fledged zero-knowledge. So far, in less than four messages, these notions

[1] Recall that a protocol is an argument if it is only computationally sound, and a proof if it is statistically sound.

[2] There are several variants of this definition strengthening/weakening different aspects [DNRS03, CLP15].

have only been obtained based on non-standard assumptions, which by now are considered implausible [BP12, BM14, BST16], for restricted classes of adversarial verifiers [BCPR14, JKKR17], or for restricted classes of distributions [FS90, Pas03]. (See the related work section for more details).

The black-box barrier. The difficulty in obtaining round-optimal zero knowledge and its relaxations stems from a fundamental barrier known as the *black-box barrier* — three-message zero-knowledge is impossible as long as the simulator is oblivious of the verifier's code, treating it as a black box [GK96]. Similar barriers hold for both weak zero-knowledge and witness hiding [HRS09]

Whereas classical zero-knowledge protocols all have black-box simulators, starting from the breakthrough work of Barak [Bar01], non-black-box techniques that exploit the verifier's code have been introduced (c.f.,[DGS09, CLP13, Goy13, BP15a, BBK$^+$16, CPS16]). However, existing techniques seem to require at least four messages (except for [BKP18], based on non-standard assumptions).

In conclusion, as in the case of zero knowledge, weak-zero-knowledge and witness-hiding protocols in three-message or less, based on standard cryptographic assumptions, remain out of reach.

1.1 Results

We devise a new non-black-box technique and apply it to obtain, under standard assumptions, weak zero-knowledge and witness hiding beyond the black-box barrier.

We now present each of our results, starting with three-message weak zero-knowledge for **NP**.

Theorem 1.1 (informal)**.** *There exists a three-message weak zero-knowledge argument for **NP** assuming: fully-homomorphic encryption, random-self-reducible encryption, two-message witness-indistinguishable arguments, non-interactive commitments, and compute-and-compare obfuscation.*

All of the above primitives are known under the LWE assumption, with the exception of fully-homomorphic encryption that also requires a circular security assumption [Gen09b, BV14, BGI$^+$17, GHKW17, GKW17, JKKR17, WZ17, BD18].[3] We can avoid the reliance on compute-and-compare obfuscation if we assume that the fully-homomorphic encryption scheme has some additional natural properties that are satisfied by known constructions (see the technical overview for more details).

We proceed to explore protocols with just one round of interaction (one verifier message followed by one prover message). While full-fledged zero-knowledge is impossible in this setting, the possibility of weak-zero knowledge and witness hiding is wide open: neither negative nor positive results are known, not even under non-standard assumptions.

Our first result in this context is a weak zero-knowledge protocol for languages in **NP** $\cap$ **coNP**.

Theorem 1.2 (informal)**.** *There exists a two-message weak zero-knowledge argument for **NP** $\cap$ **coNP** under the same assumptions as in Theorem 1.1 and non-interactive witness-indistinguishable proofs.*

This protocol gives a *natural* weak-zero-knowledge protocol that is provably not zero knowledge (assuming **NP** $\cap$ **coNP** $\not\subseteq$ **BPP**). Previously, a contrived separation was known assuming exponentially-hard injective one-way functions [CLP15]

Our next result is a two-message witness-hiding protocol for all of **NP**, assuming *witness encryption* [GGSW13]. For the time being, witness encryption is only known based on indistinguishability obfuscation, or based on non-standard assumptions on multilinear maps [GGH$^+$16, CVW18].

Theorem 1.3 (informal)**.** *There exists a two-message witness-hiding argument for **NP** under the same assumptions as in Theorem 1.1 and witness encryption.*

[3] Two-message witness-indistinguishability requires super-polynomial hardness of LWE, but can be also achieved under other standard polynomial assumptions, like factoring or the decisional linear assumption in bilinear groups. See Section 2.

The protocol we obtain is *publicly verifiable*, meaning that the proof can be verified given the transcript alone, without any secret verifier randomness. We observe that the [GO94] lower bound for two-message zero-knowledge extends also to two-message publicly-verifiable weak zero-knowledge, and thus we cannot expect to get a similar result for weak zero-knowledge.

From explainable to malicious security. The main component in all of the above results is a two-message weak-zero-knowledge argument for **NP** against a new class of verifiers that we call *explainable*. Such verifiers always choose their messages from the support of the honest verifier message distribution; namely, there exist honest verifier coins that explain their behavior. The notion resembles the notions of *semi-malicious* and *defensible* adversaries from the literature [HIK+11, BGJ+13], but differs in the fact that the verifier does not explicitly choose a random tape for the honest verifier (and it may not be possible to efficiently extract such a tape from the verifier).

Theorem 1.4 (informal)**.** *Under the same assumptions as in Theorem 1.1, there exists a two-message weak zero-knowledge argument for **NP** against explainable verifiers.*

We then give several general compilers to boost explainable security to malicious security. First, we show three-message and two-message transformations for **NP** and **NP** $\cap$ **coNP**, respectively. These transformations preserve natural notions of privacy, such as zero knowledge, weak zero-knowledge, or witness hiding, and may accordingly be of independent interest. For instance, they imply that to obtain full-fledged zero knowledge in three messages, it suffices to consider explainable verifiers. Then, we show a transformation based on witness encryption that holds for all of **NP**, but only preserves witness hiding.

1.2 Technical Overview

We now give an overview of the main ideas behind our results. We first describe how to construct two-message protocols against explainable verifiers, which is the technical core behind our results. We then describe the main ideas behind our compilers to malicious security.

Warm up: a witness-hiding protocol. Toward constructing a two-message weak zero-knowledge protocol against explainable verifiers, let us first concentrate on attend to, the easier goal of witness hiding. Recall that a protocol is witness hiding if there exists a reduction that given as input an **NP** statement x, and the code of a *witness-finding verifier*, outputs a witness. By a witness-finding verifier, we mean a verifier that given a proof that x is true, finds a witness w for x with noticeable probability.

Our protocol follows a classical paradigm by Feige, Lapidot, and Shamir [FLS99]. The first verifier message fixes a so-called *trapdoor statement* τ. In parallel, the prover and verifier execute a two-message witness-indistinguishable argument that either the statement x or the trapdoor statement τ hold. (Through the rest of the introduction we ignore the first message of the witness-indistinguishable argument.)

The trapdoor statement τ is meant to have two properties:

- To a malicious prover, trying to convince the verifier of a false statement, τ should be computationally indistinguishable from a false statement. By the soundness of the witness-indistinguishable argument, and since x is also false, the prover should fail.

- A reduction *that has the code of an explainable witness-finding verifier* should be able to obtain a witness ρ for the trapdoor statement τ. Once such a witness is found, the reduction can use it to generate the witness-indistinguishable argument. By witness indistinguishability, the reduction's proof is indistinguishable from the honestly generated proof and, therefore, the verifier will output a valid witness w for the statement x with noticeable probability.

The main challenge in executing the above paradigm is in extracting the trapdoor witness ρ from the verifier's code. The basic idea behind our non-black-box technique, and what enables such extraction, is what we call the *homomorphic trapdoor paradigm*.

In our protocol, the verifier samples a trapdoor statement τ that is indistinguishable from a false statement. On top of the trapdoor statement τ, the verifier will send an encryption ct of the witness ρ attesting that τ holds, using a fully-homomorphic encryption scheme. On one hand, by the security of the encryption scheme, this does not compromise soundness. On the other hand, a reduction that has the code of the witness-finding verifier can obtain a witness w for x *under the encryption*. To do so, the reduction homomorphically invokes the strategy described before, but under the encryption ct, where it can use ρ to compute the witness-indistinguishable argument, and obtain the witness w from the verifier.

The above step does not find a witness w in the clear (nor does it extract the trapdoor ρ). We observe, however, that an encryption of a witness w is already a non-trivial piece of information that could only be obtained given the code of the verifier and not by a malicious prover interacting with the verifier; in fact, we can use it as another trapdoor witness. Concretely, we extend our protocol to include yet another, so-called homomorphic, trapdoor statement τ_h where a witness ρ_h for τ_h could be any encryption of a witness w for x. That is, τ_h is true if and only if x is true, and a witness ρ_h for τ_h is an encryption of a witness w for x.

In the extended protocol, the prover gives a witness-indistinguishable argument that either the statement x, the trapdoor statement τ, or the homomorphic trapdoor statement τ_h hold. The reduction first uses the encrypted trapdoor ρ homomorphically to obtain a trapdoor ρ_h (in the clear), and then uses ρ_h to generate the witness-indistinguishable argument. By witness indistinguishability, the verifier will output a witness w, this time in the clear.

One difficulty in realizing the above strategy is proving the homomorphic trapdoor statement τ_h; that is, proving that there exists an encryption ρ_h of a valid witness w for x, when the reduction lacks the homomorphic decryption key. We discuss how to resolve this difficulty below when describing the more general weak-zero-knowledge protocol.

Toward weak zero-knowledge. Recall that in weak zero-knowledge, we require that there exists a simulator that given the code of the verifier and a distinguisher D, simulates the verifier's output so that it fools D. That is, D cannot ε-distinguish between the simulated output and the verifier's output in a real interaction with the prover, for any accuracy parameter ε, where the simulator is allowed to run in time polynomial in $1/\varepsilon$.

In this setting, the verifier's output is arbitrary and may not include a witness. Thus, we cannot employ the exact same strategy as before. Nevertheless, our protocol still builds on the homomorphic trapdoor paradigm, but with additional ideas. In a nutshell, instead of extracting a witness w from the verifier under the encryption, we extract a different trapdoor witness from the distinguisher D, under the encryption. Then, as before, we use the encryption of this trapdoor as the homomorphic trapdoor.

Random self-reducible encryption. To enable extraction from the distinguisher, we rely on a public-key encryption scheme that is *random self-reducible* [BM84]. In such a scheme, any distinguisher D that can tell encryptions of zero from encryptions of one with advantage ε, under some specific public key pk, can be used to decrypt arbitrary ciphertexts under the key pk, in time $\mathrm{poly}(|\mathsf{D}|/\varepsilon)$. Such schemes are known based on various standard assumptions (see Section 2.5).

The protocol. We now describe the protocol, and then go on to analyze it.

- The verifier's message, as before, includes a trapdoor statement τ and a fully-homomorphic encryption $\mathsf{ct} = \mathsf{FHE.Enc}_{\mathsf{sk}}(\rho)$ of the corresponding trapdoor witness ρ. In addition, it includes another trapdoor statement τ', and a random self-reducible encryption $\mathsf{ct}' = \mathsf{RSR.Enc}_{\mathsf{pk}}(\rho')$ of the corresponding witness ρ'. The trapdoor statements τ, τ' are both indistinguishable from false statements. This also fixes a homomorphic trapdoor statement τ_h, asserting that "there exists a fully-homomorphic encryption ρ_h of a valid witness ρ' for τ'." That is, τ_h is true if and only if τ' is true, and any witness ρ_h for τ_h is a fully-homomorphic encryption of a witness ρ' for τ'.

- The prover, as before, gives a witness-indistinguishable argument, but now in addition, it also sends a random-self-reducible encryption of zero $\mathsf{ct_P} = \mathsf{RSR.Enc_{pk}}(0)$. The witness-indistinguishable argument attests that either one of the statements x, τ, τ_h hold, or $\mathsf{ct_P}$ is an encryption of one (and not zero). Note that the trapdoor statement τ' is not directly involved in the witness-indistinguishable argument, but only defines the homomorphic trapdoor statement τ_h.

- The verifier checks that the witness-indistinguishable argument is valid and that $\mathsf{ct_P}$ decrypts to zero.

Soundness. Arguing soundness is similar to the witness-hiding protocol. Relying on the security of both encryption schemes, the verifier's encryptions of ρ and ρ' can be changed to encryptions of garbage. Then, the trapdoor statements τ and τ' can be changed to false statements, in which case the homomorphic trapdoor statement τ_h also becomes false. Also, $\mathsf{ct_P}$ must not be an encryption of one or the verifier rejects. Soundness then follows from that of the witness-indistinguishable argument.

Weak zero-knowledge. To argue weak zero-knowledge, we follow a similar approach to that taken in previous works that constructed weak zero-knowledge [BP12, JKKR17]. The simulation strategy will have two modes, a *secret mode* and a *public mode*, with two corresponding distributions on proofs, Π_s and Π_p. The secret distribution Π_s is always indistinguishable from the real distribution Π generated by the honest prover, but sampling from this distribution requires a secret s. The public distribution Π_p can be publicly sampled without knowing s. While Π_p is not indistinguishable from Π, to tell them apart, the distinguisher must "know" the secret s. That is, given any distinguisher D that ε-distinguishes Π_p from Π, it is possible to extract the secret s in time $\mathrm{poly}(|\mathsf{D}|/\varepsilon)$.

This gives rise to a simple simulation strategy that treats separately two types of distinguishers: those that know the secret s, and those that do not. Specifically, given the code of the distinguisher D and the required simulation accuracy ε, first try to extract the secret s from D, and if successful, sample from Π_s to simulate the proof. Otherwise, deduce that D cannot ε-distinguish Π_p from Π, and sample the proof from Π_p. As before, the main challenge in executing this strategy is extracting the secret s from D. Our solution again relies on the homomorphic trapdoor paradigm.

Going back to our protocol, let us define the corresponding secret and public distributions $\Pi_\mathsf{s}, \Pi_\mathsf{p}$:

- The secret distribution Π_s is associated with the homomorphic trapdoor τ_h, and can be sampled using any witness ρ_h for τ_h. Like the real proof distribution Π, it consists of an encryption of zero $\mathsf{ct_P} = \mathsf{RSR.Enc_{pk}}(0)$, but the witness-indistinguishable argument is computed using the witness ρ_h.

- The public distribution Π_p consists of an encryption of one $\mathsf{ct_P} = \mathsf{RSR.Enc_{pk}}(1)$, and the witness-indistinguishable argument is computed using the randomness of the encryption $\mathsf{ct_P}$ as a witness.

We argue that Π_s and Π_p have the required properties. The fact that Π_s is indistinguishable from the real proof distribution Π follows directly from witness indistinguishability. We now show that any distinguisher D between Π_p and Π can be used to extract a witness ρ_h (namely, an encryption of a witness ρ' for τ'), which in turn, can be used for sampling from Π_s. We do this in two steps:

1. We show that given a distinguisher D between Π_p and Π, *as well as the trapdoor* ρ, we can obtain a distinguisher D' that can tell apart encryptions of one from encryptions of zero (with about the same advantage). By random self-reducibility, such a distinguisher D' can be used to decrypt arbitrary ciphertexts under the random self-reducible scheme. In particular, such D' can be used to decrypt the encryption ct' of the trapdoor ρ' (given in the first verifier message).

   The distinguisher D' is defined in the natural way: given a bit encryption, it samples on its own a witness-indistinguishable argument, using ρ as the witness, and then applies D. By witness indistinguishability, the induced distribution Π_0, corresponding to encryptions of zero, is indistinguishable from the real proof distribution Π. Similarly, the distribution Π_1, corresponding to encryptions of one, is indistinguishable from the public distribution Π_{p}. Accordingly, any advantage of D translates to an advantage of D'.

2. In the second step, we extract the required trapdoor ρ_h allowing us to sample from Π_{s}. Analogously to the witness-hiding reduction we have already seen, this is done by applying the first step homomorphically under the encryption ct of ρ sent by the verifier. That is, under the encryption ct, we use ρ to obtain the distinguisher D' and decrypt ct'. This results in the required trapdoor ρ_h — a fully-homomorphic encryption of the trapdoor ρ'.

How to prove homomorphic trapdoor statements. To conclude our sketch of the weak zero-knowledge analysis, we explain how to prove the homomorphic trapdoor statement τ_h. As already mentioned, the difficulty in proving that there exists an encryption ρ_h of a valid witness for τ' is that the simulator does not have the corresponding secret key. Next, we discuss two possible solutions.

The first approach (which is also taken in the body of the paper) is based on obfuscation for *compute and compare programs*. A compute and compare program program $\mathbf{CC}[f, u]$ is given by a function f (represented as a circuit) and a target output string u in its range; it accepts every input x such that $f(x) = u$, and rejects all other inputs. A corresponding obfuscator compiles any such program into a program $\widetilde{\mathbf{CC}}$ with the same functionality. In terms of security, provided that the target u has high entropy , the obfuscated program is computationally indistinguishable from a simulated program that rejects all inputs. Such obfuscators are defined and constructed under LWE in [GKW17, WZ17].[4]

Using compute-and-compare obfuscation, we modify our protocol as follows. We no longer sample a trapdoor statement τ', but instead, set ρ' to be a random string. The homomorphic trapdoor statement τ_h is still defined so that a witness ρ_h is a fully-homomorphic encryption of ρ'. Specifically, τ_h is given by an obfuscation $\widetilde{\mathbf{CC}}$ of the program $\mathbf{CC}[\mathsf{FHE.Dec}_{\mathsf{sk}}, \rho']$ that accepts fully-homomorphic ciphertexts that decrypt to ρ'. Accordingly, a ciphertext ρ_h is a valid witness for τ_h if an only if $\widetilde{\mathbf{CC}}(\rho_h) = 1$. The obfuscation $\widetilde{\mathbf{CC}}$ will now be specified as part of the verifier's first message, which also includes as before the trapdoor statement τ, a fully-homomorphic encryption of the trapdoor witness ρ, and a random self-reducible encryption of the random string ρ'.

The simulator will obtain a fully-homomorphic encryption of u, which will set as a trapdoor witness ρ_h. Furthermore, as required for soundness, τ_h is indistinguishable from a false statement, since $\widetilde{\mathbf{CC}}$ is indistinguishable from a program that rejects all inputs.

A second approach. We now describe another approach for proving the trapdoor statement, which apriori seems more natural and does not rely on compute-and-compare obfuscation, but turns out to be somewhat more involved.

Here, given the fully-homomorphic encryption ρ_h of ρ', the simulator homomorphically evaluates the **NP** witness verification procedure for the statement τ' and sends the encrypted output bit. It then proves that this bit encryption was indeed obtained by homomorphically evaluating the verification procedure for τ' using the encryption ρ_h as a witness. The verifier checks the proof and in addition decrypts the output bit and checks that it is accepting.

There are some subtleties to take care of: a) to preserve witness indistinguishability, the homomorphic evaluation must be *function hiding* and b) to preserve soundness, the prover must convince the verifier that the homomorphic computation was performed over a valid ciphertext. To this end, we require a *validation* operation mapping arbitrary (possibly invalid) ciphertexts into valid ones, while preserving the plaintext underlying valid ciphertexts. Both properties can be achieved in existing fully-homomorphic encryption constructions (without additional assumptions) [Gen09a, OPP14, HW15].

[4] The known construction have a one-sided negligible correctness error. This error will not obstruct our protocol and is ignored in this introduction.

Another issue is that the simulator cannot tell whether it actually obtained the trapdoor ρ_h, or a meaningless encryption (in which case, it should deduce that D cannot tell Π from Π_{p} and use Π_{p} to simulate). This can be dealt with by running the extraction procedure again, this time in the clear, and using the candidate ρ_h as a witness, to simulate the witness-indistinguishable arguments (previously simulated using ρ under the encryption). This way, the simulator can actually obtain ρ' in the clear, and test that it is indeed a witness for τ'.

From explainable to malicious. Observe that in the protocols described above, it was crucial that the verifier behaves in an explainable fashion. In particular, the simulation (or witness-finding reduction) strongly relies on the fact that the verifier's fully-homomorphic encryption ct is indeed an encryption of trapdoor witness ρ for the statement τ. To deal with malicious adversaries, we design compilers that take a protocols secure against explainable verifiers and turn them into protocols secure against malicious verifiers. We provide three different compilers for three different settings. We now explain the main ideas behind each of these compilers.

A two-message compiler for NP $\cap$ coNP. We start by explaining our simplest compiler, which applies for languages $\mathcal{L} \in$ **NP** $\cap$ **coNP**, and preserves the message complexity of protocols with at least two messages. The compiler preserves natural security notions (like, zero-knowledge, weak zero-knowledge, or witness hiding) and relies on non-interactive witness-indistinguishable proofs.

Starting from a protocol from a protocol for proving $x \in \mathcal{L}$, the verifier in the complied protocol provides, together with every message, a non-interactive witness-indistinguishable proof that either $x \notin \mathcal{L}$ or that the verifier's messages so far "can be explained"; namely, there exists randomness for the honest verifier strategy that is consistent with the messages. Note that $x \notin \mathcal{L}$ is indeed an **NP** statement since $\mathcal{L} \in$ **coNP**.

We first argue that the complied preserves the privacy guarantee of the original protocol. By the soundness of the non-interactive proof, for every $x \in \mathcal{L}$, if the verifier sends a message that is not explainable the prover immediately aborts. Thus, the view of a malicious verifier can be simulated from that of an explainable verifier. As for soundness, if $x \notin \mathcal{L}$, then, since $\mathcal{L} \in$ **coNP**, there exists a witness for this fact. This witness can be used by a reduction to turn any cheating prover against the compiled scheme into a cheating prover against the original scheme.[5] Indeed, by witness indistinguishability, the reduction can use this witness to compute the added witness-indistinguishable proofs, without compromising the verifier's randomness.

A three-message compiler for NP. Roughly speaking, our three-message compiler for **NP** uses the extra message to map any statement x into a corresponding **coNP** statement (for some language related to $\mathcal{L}$); thereby, reducing to the previous compiler.

To explain the idea in its simplest form let us assume the existence of a perfectly binding *dense commitment scheme* where every string is a valid commitment to some value. The prover, in the first message, commits to the witness w using the dense commitment. The verifier, now proceeds as in the previous transformation, proving that the prover's commitment can be opened to a string which is *not* a valid witness for x (or that its messages can be explained).

Arguing privacy preservation is a simple extension of the argument for the two-message compiler. By the soundness of the verifier's proofs and the binding of the commitment, the verifier fails to prove that the commitment to the witness is a commitment to a non-witness. Thus, the view of a malicious verifier can be simulated given the view of an explainable verifier and a commitment to a witness. Furthermore, by the hiding of the commitment, such simulation is possible even given a commitment to garbage, which the simulator could generate alone. As for soundness, since the commitment is dense, it can be opened to *some* string, which is necessarily a non-witness, since $x \notin \mathcal{L}$. The reduction then proceeds as in the previous protocol.

[5] As is often the case when proving privacy (for instance, zero knowledge), if such a witness can only be obtained non-uniformly, the reduction would also be non-uniform.

As described, the protocol has two drawbacks. First, it relies on dense commitments, which are only known under specific assumptions, such as one-way permutations. Second, the soundness reduction is non-uniform (it requires the commitment randomness as advice). In the body, we present a variant of this compiler that avoids both problems, using an interactive proof of knowledge for the validity of the commitment.

Two-message witness-hiding compilers for NP. The above compilers do not yield a two-message protocol for languages in **NP** $\setminus$ **coNP** . Relying on witness encryption, we partially bridge this gap by giving a two-message compiler that only preserves witness-hiding.

In a witness-encryption scheme for an **NP** language $\mathcal{L}$, it is possible to encrypt messages using statements x as a public-key. Decryption can be done by anyone in possession of a corresponding witness w. In contrast, for $x \notin \mathcal{L}$, the encryption completely hides the message. The corresponding compiler is as follows. Given the statement $x \in \mathcal{L}$, the verifier sends, together with its message, an encryption of its randomness using the witness encryption scheme, and the instance x as the public-key. The honest prover, in hold of a witness, can decrypt and abort in case of malicious behavior.

The compiler guarantees that if the verifier is not explainable, the prover aborts. Therefore, intuitively, the verifier should not obtains any information. However, this intuition is misleading — a malicious verifier may generate messages without knowing whether they are explainable, and the prover's abort decision, at the least, conveys this bit of information.[6] However, a weak zero-knowledge simulator can be obtained, if it gets this information (namely, if the verifier behaved in an explainable manner) as a single bit of leakage; in particular, witness hiding is preserved — the witness-finding reduction can simply guess this bit, incurring only a constant factor decrease in its success probability.

1.3 More on Related Work

We address related work in more detail.

More on weak zero-knowledge and witness hiding. The notion of weak zero-knowledge is introduced in [DNRS03] who study the connection between 3-message *public-coin* weak zero-knowledge and so-called *magic functions*. (They also consider several variants of the definition.) The notion of witness hiding is introduced in [FS90] who prove that any witness-indistinguishable protocol is witness hiding for distributions with at least two "independent" witnesses.

The work of [CLP15] considers different notions of zero knowledge and proves that their weak variants are equivalent to their strong variants. We note that non of the resulting stronger forms is ruled out by the [GO94] two-message barrier.[7] Indeed, our results imply these forms in two messages for **NP** $\cap$ **coNP**.

Three-message constructions of weak zero-knowledge and witness hiding were shown based on auxiliary-input point obfuscation assumptions [BP12], which by now are considered implausible [BM14, BST16]. Alternatively, these notions were achieved in three-messages (under standard assumptions), and in two messages (under standard, but super-polynomial, assumptions), for a restricted class of *non-adaptive verifiers* [JKKR17]. These are verifiers who choose their message obliviously of the proven statement. In this setting, black-box simulators, or (witness-finding) reductions, are possible.

Other relaxations. Another type of relaxation considered in the literature is restricting the verifier or prover to the class of adversaries with a-priori bounded description (and arbitrary polynomial running time). Here (full-fledged) zero-knowledge can be constructed (under standard, but super-polynomial assumptions) in two messages against bounded-description verifiers and in three messages against

[6] In fact, one could come up with instantiations for witness encryption that would allow translating this information to any arbitrary predicate of the witness (see [BP15b]).

[7] All of these notions are still weak in the sense that the simulator gets any non-uniform auxiliary-input that the distinguisher does, which otherwise has to be uniform.

bounded-description provers assuming also keyless hash functions that are collision-resistant against bounded-description adversaries.

Yet another relaxation of zero-knowledge considered in the literature is that of zero-knowledge with super-polynomial simulation. These can be constructed in two messages from standard, but super-polynomial, assumptions [Pas03, BGI+17]. One-message zero-knowledge with super-polynomial simulation can be constructed against uniform provers, assuming uniform collision-resistant keyless hash functions [BP04a], or against non-uniform verifiers, but with weak soundness, assuming multi-collision-resistant keyless hash functions [BL18].

Such zero-knowledge implies a weak notion of witness hiding for distributions on instances where it is hard to find a witness, even for algorithms that run in the same super-polynomial time as the simulator.

The round complexity of zero-knowledge proofs. So far we have focused on the notion of arguments (which are only computationally sound). The round complexity of zero knowledge proofs (which are statistically sound) has also been long studied.

According to recent evidence [FGJ18], and differently from zero knowledge arguments, zero-knowledge proofs may be impossible to achieve in three messages (even with non-black-box techniques). Four-message proofs are impossible to achieve via black-box simulation, except for languages in **NP** $\cap$ **coMA** [Kat12]. Four message proofs with non-black-box simulation are only known assuming multi-collision-resistance of keyless hash functions [BKP18].

2 Preliminaries

We rely on the standard notions of Turing machines and Boolean circuits.

- We say that a Turing machine is PPT if it is probabilistic and runs in polynomial time.

- For a PPT algorithm M, we denote by $M(x;r)$ the output of M on input x and random coins r. For such an algorithm, and any input x, we may write $m \in M(x)$ to denote the fact that m is in the support of $M(x;\cdot)$.

- A polynomial-size circuit family $\mathcal{C}$ is a sequence of circuits $\mathcal{C} = \{C_\lambda\}_{\lambda\in\mathbb{N}}$, such that each circuit C_λ is of polynomial size $\lambda^{O(1)}$ and has $\lambda^{O(1)}$ input and output bits.

- We follow the standard habit of modeling any efficient adversary as a family of polynomial-size circuits. For an adversary A corresponding to a family of polynomial-size circuits $\{\mathsf{A}_\lambda\}_{\lambda\in\mathbb{N}}$, we sometimes omit the subscript λ, when it is clear from the context.

- A function $f : \mathbb{N} \to \mathbb{R}$ is negligible if $f(\lambda) = \lambda^{-\omega(1)}$ and is noticeable if $f(\lambda) = \lambda^{-O(1)}$.

- For random variables X and Y, distinguisher D, and $0 < \mu < 1$, we write $X \approx_{\mathsf{D},\mu} Y$ if

$$|\Pr[\mathsf{D}(X) = 1] - \Pr[\mathsf{D}(Y) = 1]| \leq \mu .$$

- Two ensembles of random variables $\mathcal{X} = \{X_\lambda\}_{\lambda\in\mathbb{N}}$ and $\mathcal{Y} = \{Y_\lambda\}_{\lambda\in\mathbb{N}}$ are said to be computationally indistinguishable, denoted by $\mathcal{X} \approx_c \mathcal{Y}$, if for all polynomial-size distinguishers $\mathsf{D} = \{\mathsf{D}_\lambda\}_{\lambda\in\mathbb{N}}$, there exists a negligible function μ such that for all λ,

$$X_\lambda \approx_{\mathsf{D},\mu(\lambda)} Y_\lambda \ .$$

2.1 Arguments

In what follows, we denote by $\langle \mathsf{P}, \mathsf{V} \rangle$ a protocol between two parties P and V. For input w for P, and common input x, we denote by $\mathsf{OUT}_{\mathsf{V}}\langle \mathsf{P}(w), \mathsf{V} \rangle(x)$ the output of V in the protocol. For honest verifiers, this output will be a single bit indicating acceptance (or rejection), malicious verifiers may have arbitrary output. Throughout, we assume that honest parties in all protocols are uniform PPT algorithms.

Definition 2.1 (Argument)**.** *A protocol $\langle \mathsf{P}, \mathsf{V} \rangle$ for an **NP** relation $\mathcal{R}_{\mathcal{L}}(x, w)$ is an argument if it satisfies:*

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

$$\Pr\left[\mathsf{OUT}_{\mathsf{V}}\langle \mathsf{P}(w), \mathsf{V} \rangle(x) = 1\right] = 1 \ .$$

2. **Computational soundness:** *For any polynomial-size prover $\mathsf{P}^* = \{\mathsf{P}^*_\lambda\}_\lambda$, there exists a negligible μ such that for any security parameter $\lambda \in \mathbb{N}$, and any $x \in \{0,1\}^\lambda \setminus \mathcal{L}$,*

$$\Pr\left[\mathsf{OUT}_{\mathsf{V}}\langle \mathsf{P}^*_\lambda, \mathsf{V} \rangle(x) = 1\right] \leq \mu(\lambda) \ .$$

Remark 2.1 (Delayed input). We shall also consider 2-message arguments with *delayed input*. In such arguments, the first verifier message is sampled independently of the statement, and soundness holds even for statements that are chosen adaptively (depending on the verifier message).

Remark 2.2 (Proofs). We say that the protocol is a *proof* if the soundness condition also holds against unbounded provers P^*.

Remark 2.3 (Randomized provers). We assume that provers are deterministic. As usual, this is w.l.o.g (by fixing their coins to the ones that maximize their success probability).

Definition 2.2 (Argument of knowledge)**.** *An argument $\langle \mathsf{P}, \mathsf{V} \rangle$ is an argument of knowledge if there exists a PPT extractor E, such that for any polynomial-size prover $\mathsf{P}^* = \{\mathsf{P}^*_\lambda\}_\lambda$, there exists a negligible $\mu(\lambda)$ such that for any noticeable $\varepsilon(\lambda)$, any security parameter $\lambda \in \mathbb{N}$, and any $x \in \{0,1\}^\lambda$:*

$$\begin{array}{ll} \textit{if} & \Pr\left[\mathsf{OUT}_{\mathsf{V}}\langle \mathsf{P}^*_\lambda, \mathsf{V} \rangle(x) = 1\right] \geq \varepsilon(\lambda) \ , \\ \textit{then} & \Pr\left[\begin{array}{l} w \leftarrow \mathsf{E}^{\mathsf{P}^*_\lambda}(x, 1^{1/\varepsilon}) \\ w \notin \mathcal{R}_{\mathcal{L}}(x) \end{array}\right] \leq \mu(\lambda) \ . \end{array}$$

2.1.1 Weak Zero-Knowledge

Definition 2.3 (WZK)**.** *A protocol $\langle \mathsf{P}, \mathsf{V} \rangle$ is WZK if there exists a PPT simulator S, such that for any polynomial-size verifier $\mathsf{V}^* = \{\mathsf{V}^*_\lambda\}_\lambda$ and distinguisher $\mathsf{D} = \{\mathsf{D}_\lambda\}_\lambda$, and noticeable $\varepsilon(\lambda)$, there exists a negligible $\mu(\lambda)$, such that for any $\lambda \in \mathbb{N}$ and $x \in \mathcal{L} \cap \{0,1\}^\lambda$,*

$$\mathsf{OUT}_{\mathsf{V}^*_\lambda}\langle \mathsf{P}(w), \mathsf{V}^*_\lambda \rangle(x) \approx_{\mathsf{D}_\lambda, \varepsilon+\mu} \mathsf{S}(x, \mathsf{V}^*_\lambda, \mathsf{D}_\lambda, 1^{1/\varepsilon}) \ .$$

Remark 2.4 (Randomized adversaries). Above we assume that V^* and D are deterministic. As in the case of (standard) zero knowledge with universal simulator, this is known to be w.l.o.g (it is equivalent to a definition for randomized V^* and D, as the above definition would hold for *any* hardwired randomness).

Throughout the paper, we may think about V and D being either deterministic or randomized.

2.1.2 Witness Hiding

Definition 2.4 (WH)**.** *A protocol $\langle \mathsf{P}, \mathsf{V} \rangle$ is WH if there exists a PPT reduction R, such that for any polynomial-size verifier $\mathsf{V}^* = \{\mathsf{V}^*_\lambda\}_\lambda$ and noticeable $\varepsilon(\lambda)$, there exists a negligible $\mu(\lambda)$, such that for any $\lambda \in \mathbb{N}$ and $x \in \mathcal{L} \cap \{0,1\}^\lambda$,*

$$\Pr\left[\mathsf{OUT}_{\mathsf{V}^*_\lambda}\langle \mathsf{P}(w), \mathsf{V}^*_\lambda \rangle(x) \in \mathcal{R}_\mathcal{L}(x)\right] \leq \Pr\left[\mathsf{R}(x, \mathsf{V}^*_\lambda, 1^{1/\varepsilon}) \in \mathcal{R}_\mathcal{L}(x)\right] + \varepsilon(\lambda) + \mu(\lambda) \enspace .$$

Remark 2.5 (Randomized verifiers). As in Remark 2.4, and for the same reasons, the above definition considers w.l.o.g only deterministic verifiers V^*.

It is well-known that WZK implies WH (by considering the specific distinguishers that checks if the verifier's output is a valid witness).

Lemma 2.1. *Any WZK protocol is WH.*

2.1.3 Witness Indistinguishability

Definition 2.5 (WI)**.** *A protocol $\langle \mathsf{P}, \mathsf{V} \rangle$ is WI if for any polynomial-size verifier $\mathsf{V}^* = \{\mathsf{V}^*_\lambda\}_\lambda$ and distinguisher $\mathsf{D} = \{\mathsf{D}_\lambda\}_\lambda$, there exists a negligible $\mu(\lambda)$, such that for any $\lambda \in \mathbb{N}$, $x \in \mathcal{L} \cap \{0,1\}^\lambda$, $w_0, w_1 \in \mathcal{R}_\mathcal{L}(x)$,*

$$\mathsf{OUT}_{\mathsf{V}^*_\lambda}\langle \mathsf{P}(w_0), \mathsf{V}^*_\lambda \rangle(x) \approx_{\mathsf{D}_\lambda, \mu} \mathsf{OUT}_{\mathsf{V}^*_\lambda}\langle \mathsf{P}(w_1), \mathsf{V}^*_\lambda \rangle(x) \enspace .$$

Instantiations. Each of our constructions relies on (some of) the following three types of WI systems:

- A 3-message WI argument of knowledge. Such arguments are known based on any non-interactive commitment [GMW91] (in particular, based on LWE[GHKW17]).

- A 2-message WI argument with delayed input. Such arguments that are publicly verifiable (and in fact also proofs) can be based either on trapdoor permutations [DN07], standard assumptions on bilinear groups [GOS12], or indistinguishability obfuscation [BP15b].

  Such privately-verifiable arguments can be constructed from any 2-message oblivious transfer against malicious receivers and super-polynomial semi-honest senders [BGI+17, JKKR17]. In particular, they can be constructed from (super-polynomial) LWE [BD18].

- Non-interactive WI (NIWI) proofs. Such proofs are know based on trapdoor permutations and derandomization assumptions [BOV07], standard assumptions on bilinear groups [GOS12], or indistinguishability obfuscation and one-way permutations [BP15b].

2.1.4 Explainable Verifiers

Let $\langle \mathsf{P}, \mathsf{V} \rangle$ be a protocol. Let $\mathsf{HOUT}_{\mathsf{V}^*}\langle \mathsf{P}(w), \mathsf{V}^* \rangle(x)$ denote the honest-truncation output of V^* in the protocol $\langle \mathsf{P}, \mathsf{V} \rangle$, which equals:

- $\mathsf{OUT}_{\mathsf{V}^*}\langle \mathsf{P}(w), \mathsf{V}^* \rangle(x)$, the actual output, if all messages sent by V^* are in the support of the honest V; namely, there exists a random tape for the honest verifier, which explains these messages.

- $\perp$ otherwise.

Definition 2.6 (explainable verifier)**.** *We say that $\mathsf{V}^* = \{\mathsf{V}^*_\lambda\}_\lambda$ is explainable in a protocol $\langle \mathsf{P}, \mathsf{V} \rangle$ if for any (x, w),*

$$\Pr\left[\mathsf{HOUT}_{\mathsf{V}^*_\lambda}\langle \mathsf{P}(w), \mathsf{V}^*_\lambda \rangle(x) = \mathsf{OUT}_{\mathsf{V}^*_\lambda}\langle \mathsf{P}(w), \mathsf{V}^*_\lambda \rangle(x)\right] = 1 - \lambda^{-\omega(1)} \enspace .$$

Explainable WZK and WH. WZK and WH against explainable verifiers are defined exactly as WZK and WH only that the (respective) definition only holds against *explainable* verifiers rather than *all* verifiers.

We also note that Lemma 2.1 saying that WZK implies WH also holds for explainable verifiers.

Remark 2.6 (Randomized verifiers). For randomized verifies, we naturally define explainable behavior consistently with Definition 2.6, only that the probability of explainable behavior is also taken over their own coins (and not just the prover's).

As in Remark 2.4, for both WZK and WH, explainable security against deterministic adversaries implies explainable security against randomized adversaries.

Weak WH. We will also consider a notion of weak WH that does not restrict the verifier to be explainable, but only requires that the witness-finding reduction finds a witness with the same probability that the verifier finds a witness *and is explainable*.

Definition 2.7 (Weak WH). *A protocol* $\langle \mathsf{P}, \mathsf{V} \rangle$ *is weakly WH if there exists a PPT reduction* R*, such that for any polynomial-size verifier* $\mathsf{V}^* = \{\mathsf{V}^*_\lambda\}_\lambda$ *and noticeable* $\varepsilon(\lambda)$*, there exists a negligible* $\mu(\lambda)$*, such that for any* $\lambda \in \mathbb{N}$ *and* $x \in \mathcal{L} \cap \{0,1\}^\lambda$*,*

$$\Pr\left[\mathsf{HOUT}_{\mathsf{V}^*_\lambda}\langle \mathsf{P}(w), \mathsf{V}^*_\lambda \rangle(x) \in \mathcal{R}_\mathcal{L}(x)\right] \le \Pr\left[\mathsf{R}(x, \mathsf{V}^*_\lambda, 1^{1/\varepsilon}) \in \mathcal{R}_\mathcal{L}(x)\right] + \varepsilon(\lambda) + \mu(\lambda) \ .$$

Lemma 2.2. *Any two-message WH protocol against explainable verifiers is also weakly WH (against arbitrary verifiers).*

Proof. Let R be the witness-finding reduction for an explainable two-message WH $\langle \mathsf{P}, \mathsf{V} \rangle$, we describe a variant R' of R that establishes that the system is weakly WH.

$\mathsf{R}'(x, \mathsf{V}^*_\lambda, 1^{1/\varepsilon})$ first constructs a new verifier V^*_x that given x acts exactly as V^*_λ, and for any other x acts honestly according to V and outputs $\perp$. It then runs $\mathsf{R}(x, \mathsf{V}^*_x, 1^{1/\varepsilon})$.

We now argue that R' satisfies the weak WH property. Fix any polynomial-size $\mathsf{V}^* = \{\mathsf{V}^*_\lambda\}_\lambda$, and assume toward contradiction that there exist noticeable $\varepsilon(\lambda), \delta(\lambda)$ such that for infinitely many $\lambda \in \mathbb{N}$ and $x_\lambda \in \{0,1\}^\lambda \cap \mathcal{L}$ and $w_\lambda \in \mathcal{R}_\mathcal{L}(x_\lambda)$, it holds that

$$\Pr\left[\mathsf{HOUT}_{\mathsf{V}^*_\lambda}\langle \mathsf{P}(w_\lambda), \mathsf{V}^*_\lambda \rangle(x_\lambda) \in \mathcal{R}_\mathcal{L}(x_\lambda)\right] > \Pr\left[\mathsf{R}'(x_\lambda, \mathsf{V}^*_\lambda, 1^{1/\varepsilon}) \in \mathcal{R}_\mathcal{L}(x_\lambda)\right] + \varepsilon(\lambda) + \delta(\lambda) \ .$$

Then, in particular, for any such x_λ, V^*_λ sends an explainable message. This means that the verifier $\{\mathsf{V}^*_{x_\lambda}\}_\lambda$, as defined above, is explainable.

Furthermore, by the above equation, and the definition of R',

$$\Pr\left[\mathsf{OUT}_{\mathsf{V}^*_{x_\lambda}}\langle \mathsf{P}(w_\lambda), \mathsf{V}^*_{x_\lambda} \rangle(x_\lambda) \in \mathcal{R}_\mathcal{L}(x_\lambda)\right] > \Pr\left[\mathsf{R}(x_\lambda, \mathsf{V}^*_{x_\lambda}, 1^{1/\varepsilon}) \in \mathcal{R}_\mathcal{L}(x_\lambda)\right] + \varepsilon(\lambda) + \delta(\lambda) \ ,$$

which contradicts WH against explainable verifiers. □

2.2 Bit Commitments

We define (non-interactive) bit commitments.

Definition 2.8 (Bit commitment). *A polynomial-time computable function*

$$\mathsf{Com} : \{0,1\} \times \{0,1\}^* \to \{0,1\}^*$$

is a bit commitment if it satisfies:

1. **Binding:** *For any* $r, r' \in \{0,1\}^*, b, b' \in \{0,1\}$*, if* $\mathsf{Com}(b; r) = \mathsf{Com}(b'; r')$ *then* $b = b'$*.*

2. **Computational hiding:** *For any polynomial-size distinguisher* $\mathsf{D} = \{\mathsf{D}_\lambda\}_{\lambda\in\mathbb{N}}$, *there exists a negligible* μ *such that for any security parameter* $\lambda \in \mathbb{N}$,

$$\mathsf{Com}(0) \approx_{\mathsf{D}_\lambda,\mu} \mathsf{Com}(1) \ ,$$

where $\mathsf{Com}(b)$ *is the distribution of commitments to* b *with randomness* $r \leftarrow \{0,1\}^\lambda$.

Instantiations. (Non-interactive) bit commitments are known based on various standard assumptions, including LWE [GHKW17].

2.3 Fully-Homomorphic Encryption

We recall the definition of fully-homomorphic encryption (FHE).

Definition 2.9. *A fully-homomorphic encryption scheme* $(\mathsf{FHE.Enc}, \mathsf{FHE.Dec}, \mathsf{FHE.Eval})$ *satisfies*

1. **Correctness:** *for any* $\lambda \in \mathbb{N}, \mathsf{sk} \in \{0,1\}^\lambda$, *message* $m \in \{0,1\}^*$, *and circuit* C,

$$\mathsf{FHE.Dec}_{\mathsf{sk}}(\mathsf{FHE.Eval}(C, \mathsf{FHE.Enc}_{\mathsf{sk}}(m))) = C(m) \ .$$

2. **Indistinguishability:** *For any polynomial-size distinguisher* $\mathsf{D} = \{\mathsf{D}_\lambda\}_{\lambda\in\mathbb{N}}$, *there exists a negligible* μ *such that for any security parameter* $\lambda \in \mathbb{N}$, *and any two equal-length messages* m_0, m_1,

$$\mathsf{FHE.Enc}_{\mathsf{sk}}(m_0) \approx_{\mathsf{D}_\lambda,\mu} \mathsf{FHE.Enc}_{\mathsf{sk}}(m_1) \ ,$$

where $\mathsf{FHE.Enc}_{\mathsf{sk}}(m_b)$ *is the distribution of encryptions of* m_b *with random secret key* $\mathsf{sk} \leftarrow \{0,1\}^\lambda$.

Instantiations. Starting from the work of Gentry [Gen09a], there have been several constructions of FHE schemes, including ones based on LWE and a corresponding circular security assumptions, starting from [BV14].[8]

2.4 Compute and Compare Obfuscation

We start by defining the class of *compute and compare programs.*

Definition 2.10 (Compute and compare)**.** *Let* $f : \{0,1\}^n \to \{0,1\}^\lambda$ *be a circuit, and let* $u \in \{0,1\}^\lambda$ *be a string. Then* $\mathbf{CC}[f,u](x)$ *is a circuit that returns* 1 *if* $f(x) = y$, *and* 0 *otherwise.*

We now define compute and compare (CC) obfuscators. In what follows $\mathcal{O}$ is a PPT algorithm that takes as input a CC circuit $\mathbf{CC}[f,u]$ and outputs a new circuit $\widetilde{\mathbf{CC}}$. (We assume that the CC circuit $\mathbf{CC}[f,u]$ is given in some canonical description from which f and u can be read.)

Definition 2.11 (CC obfuscator)**.** *A PPT* $\mathcal{O}$ *is a compute and compare obfuscator if it satisfies:*

1. **One-sided correctness:** *for any circuit* $f : \{0,1\}^n \to \{0,1\}^\lambda$ *and* $u \in \{0,1\}^\lambda$, *and any* $x \in \{0,1\}^n$ *such that* $f(x) = u$,

$$\Pr\left[\widetilde{\mathbf{CC}}(x) = 1 \;\middle|\; \widetilde{\mathbf{CC}} \leftarrow \mathcal{O}(\mathbf{CC}[f,u])\right] = 1 \ .$$

2. **Simulation:** *there exists a PPT simulator* Sim *such that*

[8] While leveled FHE is known based on LWE alone (without circuit security), it will not be sufficient for this work.

- *For any polynomially-bounded function $\ell(\lambda)$ and any polynomial-size distinguisher $\mathsf{D} = \{\mathsf{D}_\lambda\}_{\lambda\in\mathbb{N}}$, there exists a negligible μ such that for any $\lambda \in \mathbb{N}$ and $\ell(\lambda)$-size circuit $f : \{0,1\}^n \to \{0,1\}^\lambda$,*
$$\mathcal{O}(\mathbf{CC}[f,u]) \approx_{\mathsf{D}_\lambda,\mu} \mathsf{Sim}(1^\lambda, 1^\ell) \ ,$$
where $u \leftarrow \{0,1\}^\lambda$ is chosen uniformly at random.
- *Simulated circuits are rejecting:*
$$\Pr\left[\exists x : \widetilde{\mathbf{CC}}(x) = 1 \,\middle|\, \widetilde{\mathbf{CC}} \leftarrow \mathsf{Sim}(1^\lambda, 1^\ell)\right] \le \lambda^{-\omega(1)} \ .$$

Instantiations. Compute and compare obfuscators are constructed in [GKW17, WZ17] based on LWE.

The correctness considered there is two-sided — they prove perfect correctness for inputs x such that $f(x) = u$ and almost perfect correctness for inputs x such that $f(x) \neq u$. We will only rely on the first of the two (and perfect correctness will play a role).

In addition, they do not state explicitly the fact that simulated circuits are rejecting. However, this is satisfied by their construction, and follows readily from their simulator definition (see e.g., [WZ17, Claim 4.11]) and correctness analysis (see e.g., [WZ17, Claim 4.11]).

2.5 Random Self-Reducible Public-Key Encryption

Intuitively speaking, a *random self reducible* (RSR) encryption scheme admits the classical notion of random self-reduction [BM84] — it is possible to rerandomize an arbitrary ciphertext into a random ciphertexts of the same message under the same public key. More generally, given access to an average-case distinguisher, it is possible to decrypt in the worst case.

Syntax. An RSR encryption scheme RSR consists of PPT algorithms $(\mathsf{RSR.Gen}, \mathsf{RSR.Enc}, \mathsf{RSR.Dec}, \mathsf{RSR}.\widetilde{\mathsf{Dec}})$. The first three algorithms have the standard syntax of a public-key (bit) encryption scheme.

The fourth algorithm $\mathsf{RSR}.\widetilde{\mathsf{Dec}}^{\mathsf{D}}(\mathsf{ct}, \mathsf{pk}, 1^{1/\varepsilon})$ is an oracle-aided alternative decryption algorithm, which given as input a ciphertext ct, public key pk and (distinguishing) parameter $1^{1/\varepsilon}$, and a oracle access to a distinguisher D, outputs a plaintext bit b.

Definition 2.12. *A public-key encryption scheme* $(\mathsf{RSR.Gen}, \mathsf{RSR.Enc}, \mathsf{RSR.Dec}, \mathsf{RSR}.\widetilde{\mathsf{Dec}})$ *is random self-reducible if it satisfies*

1. **Correctness:** *for any $b \in \{0,1\}, \lambda \in \mathbb{N}$,*
$$\Pr\left[\mathsf{RSR.Dec}_{\mathsf{sk}}(\mathsf{ct}) = b \,\middle|\, \begin{array}{c}(\mathsf{sk},\mathsf{pk}) \leftarrow \mathsf{RSR.Gen}(1^\lambda) \\ \mathsf{ct} \leftarrow \mathsf{RSR.Enc}_{\mathsf{pk}}(b)\end{array}\right] = 1$$

2. **Indistinguishability:** *For any polynomial-size distinguisher $\mathsf{D} = \{\mathsf{D}_\lambda\}_{\lambda\in\mathbb{N}}$, there exists a negligible μ such that for any security parameter $\lambda \in \mathbb{N}$,*
$$\mathsf{RSR.Enc}_{\mathsf{pk}}(0) \approx_{\mathsf{D}_\lambda,\mu} \mathsf{RSR.Enc}_{\mathsf{pk}}(1) \ ,$$
where $\mathsf{RSR.Enc}_{\mathsf{pk}}(b)$ is the distribution of encryptions of b with random public key $\mathsf{pk} \leftarrow \mathsf{RSR.Gen}(1^\lambda)$.

3. **Random self-reduction:** *for any public key $\mathsf{pk} \in \mathsf{RSR.Gen}(1^\lambda)$ it holds that for any distinguisher D and ε,*
   - *if*
$$|\mathbb{E}\mathsf{D}(\mathsf{RSR.Enc}_{\mathsf{pk}}(0)) - \mathbb{E}\mathsf{D}(\mathsf{RSR.Enc}_{\mathsf{pk}}(1))| \ge \varepsilon \ ,$$

- *then for any* $b \in \{0,1\}$ *and* $\mathsf{ct} \in \mathsf{RSR.Enc}_{\mathsf{pk}}(b)$,

  $$\Pr\left[\mathsf{RSR}.\widetilde{\mathsf{Dec}}^{\mathsf{D}}(\mathsf{ct}, \mathsf{pk}, 1^{1/\varepsilon}) = b\right] = 1 - \lambda^{-\omega(1)} \ ,$$

  where the probability is over the coins of $\mathsf{RSR}.\widetilde{\mathsf{Dec}}$.

Relaxed RSR. We may consider a relaxed version of RSR encryption, where ciphertexts $\mathsf{ct} \in \mathsf{RSR.Enc}_{\mathsf{pk}}(b)$ can be reduced to ciphertexts relative to a different encryption algorithm $\mathsf{RSR}.\widetilde{\mathsf{Enc}}$. Such a relaxation is simpler to construct for instance under LWE.

Formally, there exists an additional PPT algorithm $\mathsf{RSR}.\widetilde{\mathsf{Enc}}$, that satisfies:

1. **Correctness:** similarly to $\mathsf{RSR.Enc}$.
2. **Relaxed random self-reduction:** for any public key $\mathsf{pk} \in \mathsf{RSR.Gen}(1^\lambda)$ it holds that for any distinguisher D and ε,
   - if
     $$\left|\mathbb{E}\mathsf{D}(\mathsf{RSR}.\widetilde{\mathsf{Enc}}_{\mathsf{pk}}(0)) - \mathbb{E}\mathsf{D}(\mathsf{RSR}.\widetilde{\mathsf{Enc}}_{\mathsf{pk}}(1))\right| \geq \varepsilon \ ,$$
   - then for any $b \in \{0,1\}$ and $\mathsf{ct} \in \mathsf{RSR.Enc}_{\mathsf{pk}}(b)$,
     $$\Pr\left[\mathsf{RSR}.\widetilde{\mathsf{Dec}}^{\mathsf{D}}(\mathsf{ct}, \mathsf{pk}, 1^{1/\varepsilon}) = b\right] = 1 - \lambda^{-\omega(1)} \ ,$$
     where the probability is over the coins of $\mathsf{RSR}.\widetilde{\mathsf{Dec}}$.

We do not explicitly define (nor use) semantic security for the alternative encryption algorithm (although it actually follows from the semantic security of the original encryption together with relaxed RSR).

Instantiations. There are various public-key encryption schemes [GM84, Gam85, Pai99] based on standard algebraic assumptions, that are known to be perfectly *rerandomizable* and are hence random self reducible.

Statistically rerandomizable schemes are also known based on LWE [Reg09]. However, in such schemes rerandomization is guaranteed for a random public key, whereas as we require that it holds for an arbitrary public key in the support of the generation algorithm. LWE does easily give relaxed RSR schemes using the standard noise flooding technique [Gen09a].

2.6 Witness Encryption

We recall the definition of witness encryption (WE).

Definition 2.13. *A witness encryption scheme* $(\mathsf{WE.Enc}, \mathsf{WE.Dec})$ *for an* ***NP*** *language* $\mathcal{L}$ *satisfies*

1. **Correctness:** *for any* $(x, w) \in \mathcal{R}_{\mathcal{L}}$*, and message* $m \in \{0,1\}^*$,
   $$\mathsf{WE.Dec}_w(\mathsf{WE.Enc}_x(m))) = m \ .$$
2. **Indistinguishability:** *For any polynomial-size distinguisher* $\mathsf{D} = \{\mathsf{D}_\lambda\}_{\lambda \in \mathbb{N}}$*, there exists a negligible* μ *such that for any security parameter* $\lambda \in \mathbb{N}$*, any* $x \in \{0,1\}^\lambda \setminus \mathcal{L}$*, and two equal-length messages* m_0, m_1,
   $$\mathsf{WE.Enc}_x(m_0) \approx_{\mathsf{D}_\lambda, \mu} \mathsf{WE.Enc}_x(m_1) \ ,$$
   where $\mathsf{WE.Enc}_x(m_b)$ *is the distribution of encryptions of* m_b *under* x.

Instantiations. Starting from the work of Garg, Gentry, Sahai, and Waters [GGSW13], there have been several constructions of WE schemes. State of the art schemes (which haven't been broken) includes constructions based on indistinguishability obfuscation [GGH⁺16] or the GGH15 multi-linear maps [CVW18].

3 Weak Zero-Knowledge against Explainable Verifiers

In this section, we construct a two-message WZK argument against explainable verifiers. We start by presenting the protocol and then proceed to analyze it.

Ingredients and notation:

- A 2-message WI argument for **NP** with delayed input. We denote its messages by $(\mathsf{wi}_1, \mathsf{wi}_2)$.
- A non-interactive perfectly-binding commitment scheme Com.
- A fully-homomorphic encryption scheme $(\mathsf{FHE.Enc}, \mathsf{FHE.Dec}, \mathsf{FHE.Eval})$.
- A compute-and-compare obfuscator $\mathcal{O}$.
- A random self-reducible public-key encryption $(\mathsf{RSR.Gen}, \mathsf{RSR.Enc}, \mathsf{RSR.Dec}, \mathsf{RSR.}\widetilde{\mathsf{Dec}})$.
  (In fact, relaxed RSR suffices. To simplify the description of the protocol, we rely on standard RSR, and later remark why relaxed RSR suffices.)

We describe the protocol in Figure 1.

Protocol 1

Common Input: an instance $x \in \mathcal{L} \cap \{0,1\}^\lambda$, for security parameter λ.

P**'s auxiliary input:** a witness $w \in \mathcal{R}_\mathcal{L}(x)$.

1. V computes
   - wi_1, the first message of the WI argument,
   - $\mathsf{cmt} \leftarrow \mathsf{Com}(0; r)$, a commitment to zero, using randomness $r \leftarrow \{0,1\}^\lambda$,
   - $\mathsf{ct} \leftarrow \mathsf{FHE.Enc}_{\mathsf{sk}}(r)$, an encryption of the commitment randomness, under a randomly chosen secret key $\mathsf{sk} \leftarrow \{0,1\}^\lambda$.
   - $\widetilde{\mathbf{CC}} \leftarrow \mathcal{O}(\mathbf{CC}[\mathsf{FHE.Dec}_{\mathsf{sk}}, u])$, an obfuscation of the CC program given by the FHE decryption circuit and a random target $u \leftarrow \{0,1\}^\lambda$.
   - $\mathsf{ct}' \leftarrow \mathsf{RSR.Enc}_{\mathsf{pk}}(u)$, an RSR encryption of the target u, where $(\mathsf{sk}', \mathsf{pk}') \leftarrow \mathsf{RSR.Gen}(1^\lambda)$ are randomly chosen keys.

   It sends $(\mathsf{wi}_1, \mathsf{cmt}, \mathsf{ct}, \widetilde{\mathbf{CC}}, \mathsf{ct}', \mathsf{pk}')$.

2. P computes
   - $\mathsf{ct}'' \leftarrow \mathsf{RSR.Enc}_{\mathsf{pk}'}(1)$, an RSR (bit-by-bit) encryption of 1.
   - wi_2, the second WI message for the statement $\Psi(x, \mathsf{cmt}, \widetilde{\mathbf{CC}}, \mathsf{ct}'', \mathsf{pk}')$ given by:

   $$\begin{array}{lcll} \exists w & : & (x, w) \in \mathcal{R}_\mathcal{L} & \bigvee \\ \exists r & : & \mathsf{cmt} = \mathsf{Com}(0; r) & \bigvee \\ \exists \widehat{\mathsf{ct}} & : & \widetilde{\mathbf{CC}}(\widehat{\mathsf{ct}}) = 1 & \bigvee \\ \exists r' & : & \mathsf{ct}'' = \mathsf{RSR.Enc}_{\mathsf{pk}'}(0; r') \quad , & \end{array}$$

   using the witness $w \in \mathcal{R}_\mathcal{L}(x)$.

   It sends $(\mathsf{ct}'', \mathsf{wi}_2)$.

3. V verifies
   - the WI argument $(\mathsf{wi}_1, \mathsf{wi}_2)$ for the statement Ψ,
   - that $\mathsf{RSR}_{\mathsf{sk}'}(\mathsf{ct}'') = 1$.

Figure 1: A 2-message WZK argument $\langle \mathsf{P}, \mathsf{V} \rangle$ for **NP**.

3.1 Analysis

We now analyze the protocol. We first show that it is sound, and then that it is WZK against explainable verifiers.

Proposition 3.1. *Protocol 1 is sound.*

Proof. To prove soundness, we consider several hybrid protocols, and show that the probability that a malicious prover can cheat is preserved throughout the hybrids.

$\mathcal{H}_0$**:** This is the real protocol.

$\mathcal{H}_1$**:** In this hybrid, ct' is an encryption of 0^λ instead of the target u.

This hybrid is indistinguishable from the previous one by the semantic security of the RSR scheme RSR.

$\mathcal{H}_2$**:** In this hybrid, the obfuscation $\widetilde{\mathbf{CC}} \leftarrow \mathsf{Sim}(1^\lambda, 1^\ell)$ is simulated rather than an obfuscation of $\mathbf{CC}[\mathsf{FHE.Dec}_{\mathsf{sk}}, u]$, where ℓ is the size of the decryption circuit $\mathsf{FHE.Dec}_{\mathsf{sk}}$.

This hybrid is indistinguishable from the previous one by the CC simulation guarantee; indeed, in the previous hybrid, the target u is uniformly random and independent of $\mathsf{FHE.Dec}_{\mathsf{sk}}$, and the rest of the experiment.

$\mathcal{H}_3$**:** In this hybrid, ct is an encryption of 0^λ instead of the commitment randomness r.

This hybrid is indistinguishable from the previous one by the semantic security of the FHE scheme FHE.

$\mathcal{H}_4$**:** In this hybrid, cmt is a commitment to 1 instead of 0.

This hybrid is indistinguishable from the previous one by hiding of the commitment.

It is left to show that in $\mathcal{H}_4$, no malicious prover can convince the verifier to accept a false statement $x \notin \mathcal{L}$, except with negligible probability.

Observe that in this hybrid:

- $x \notin \mathcal{L}$.
- $\mathsf{cmt} \in \mathsf{Com}(1)$, and by perfect binding there does not exist r such that $\mathsf{cmt} = \mathsf{Com}(0; r)$.
- By the CC simulation guarantee, except with negligible probability, there does not exist $\widehat{\mathsf{ct}}$ such that $\widetilde{\mathbf{CC}}(\widehat{\mathsf{ct}}) = 1$.
- By the definition of the verifier and the perfect correctness of the encryption scheme RSR, if the verifier accepts, there does not exist r' such that $\mathsf{ct}'' = \mathsf{RSR.Enc}(0; r)$.

It follows that the verifier accepts only if it accepts a WI argument to a false statement $\Psi(x, \mathsf{cmt}, \widetilde{\mathbf{CC}}, \mathsf{ct}'', \mathsf{pk}')$. Soundness now follows by the soundness of the WI argument. □

Proposition 3.2. *Protocol 1 is weak zero-knowledge against explainable verifiers.*

Proof. We describe the simulator S. Throughout, we assume w.l.o.g that the output of V^* consists of the prover message. (Otherwise, we can consider a new verifier of this form along with a new distinguisher who computes internally the original verifier output, and then applies the original distinguisher.)

$\mathsf{S}(x, \mathsf{V}^*_\lambda, \mathsf{D}_\lambda, 1^{1/\varepsilon})$**:**

- Obtains $(\mathsf{wi}_1, \mathsf{cmt}, \mathsf{ct}, \widetilde{\mathbf{CC}}, \mathsf{ct}', \mathsf{pk}')$ from V^*_λ.
- Constructs the (*homomorphic simulation*) circuit $\mathsf{HS}(r)$ that given an input r:
  - Constructs a distinguisher $\mathsf{D}'(\mathsf{ct}'')$ for the RSR encryption scheme that given a ciphertext ct'':
    - $*$ Samples a second WI message wi_2 for the statement $\Psi(x, \mathsf{cmt}, \widetilde{\mathbf{CC}}, \mathsf{ct}'', \mathsf{pk}')$, using as the witness the randomness r attesting that $\mathsf{cmt} = \mathsf{Com}(0; r)$.
    - $*$ Runs $\mathsf{D}_\lambda(\mathsf{ct}'', \mathsf{wi}_2)$.
  - Apply the decryptor

$$\tilde{u} \leftarrow \mathsf{RSR}.\widetilde{\mathsf{Dec}}^{\mathsf{D}'}(\mathsf{ct}', \mathsf{pk}', 1^{1/\varepsilon}) \enspace ,$$

and output $\tilde{u}$.

All randomness required by the above is hardwired to HS.

- Compute $\widehat{\mathsf{ct}} = \mathsf{FHE.Eval}(\mathsf{HS}, \mathsf{ct})$.
- If $\widetilde{\mathbf{CC}}(\widehat{\mathsf{ct}}) = 1$:
  - Sample $\mathsf{ct}'' \leftarrow \mathsf{RSR.Enc}_{\mathsf{pk}'}(1)$.
  - Sample a second WI message wi_2 for the statement $\Psi(x, \mathsf{cmt}, \widetilde{\mathbf{CC}}, \mathsf{ct}'', \mathsf{pk}')$, using as the witness $\widehat{\mathsf{ct}}$ attesting that $\widetilde{\mathbf{CC}}(\widehat{\mathsf{ct}}) = 1$.
- Otherwise:
  - Sample $\mathsf{ct}'' \leftarrow \mathsf{RSR.Enc}_{\mathsf{pk}'}(0)$.
  - Sample a second WI message wi_2 for the statement $\Psi(x, \mathsf{cmt}, \widetilde{\mathbf{CC}}, \mathsf{ct}'', \mathsf{pk}')$, using as the witness the randomness r' attesting that $\mathsf{ct}'' \leftarrow \mathsf{RSR.Enc}_{\mathsf{pk}'}(0; r')$.
  -
- Output $(\mathsf{ct}'', \mathsf{wi}_2)$.

Simulation validity. The simulator clearly runs in polynomial time (in its input length). We focus on proving validity.

To prove that validity of the simulator, let $\mathsf{V}^* = \{\mathsf{V}^*_\lambda\}_\lambda$ be a polynomial-size explainable verifier, let $\mathsf{D} = \{\mathsf{D}_\lambda\}_\lambda$ be a polynomial-size distinguisher, and let $\varepsilon(\lambda) = \lambda^{-O(1)}$. Fix any λ, and $x \in \mathcal{L} \cap \{0,1\}^\lambda$ such that the message sent by V^*_λ is explainable (i.e., it is consistent with some honest verifier message as specified in Protocol 1).

Let Δ be the advantage of the distinguisher D' (as defined in the simulation procedure) in distinguishing zero-encryptions from one-encryptions:

$$\Delta := \left| \mathop{\mathbb{E}}_{\mathsf{D}', \mathsf{RSR.Enc}} \left[\mathsf{D}'(\mathsf{RSR.Enc}_{\mathsf{pk}'}(0)) - \mathsf{D}'(\mathsf{RSR.Enc}_{\mathsf{pk}'}(1)) \right] \right| \ .$$

We consider two cases.

Case 1: $\Delta > \varepsilon$. Let u be the target string underlying the obfuscated CC program $\widetilde{\mathbf{CC}}$, and recall that $\mathsf{ct}' \in \mathsf{RSR.Enc}_{\mathsf{pk}'}(u)$. Then by the random self reducibility of RSR, with overwhelming probability $1 - \lambda^{-\omega(1)}$, it holds that

$$\mathsf{RSR}.\widetilde{\mathsf{Dec}}^{\mathsf{D}'}(\mathsf{ct}', \mathsf{pk}', 1^{1/\varepsilon}) = u \ ,$$

in which case, $\mathsf{HS}(r) = u$. From hereon, we assume that this is the case.

By the correctness of FHE, the ciphertext $\widehat{\mathsf{ct}} = \mathsf{FHE.Eval}(\mathsf{HS}, \widehat{\mathsf{ct}})$ obtained by the simulator satisfies:

$$\mathsf{FHE.Dec}_{\mathsf{sk}}(\widehat{\mathsf{ct}}) = u \ ,$$

and thus by the one-sided correctness of the CC obfuscator $\mathcal{O}$,

$$\widetilde{\mathbf{CC}}(\widehat{\mathsf{ct}}) = 1 \ .$$

It follows that in this case, except with negligible probability $\lambda^{-\omega(1)}$, the simulator generates a prover message $(\mathsf{ct}'', \mathsf{wi}_2)$ exactly like the honest prover, with the exception of using $\widehat{\mathsf{ct}}$ as the witness for computing wi_2, instead of $w \in \mathcal{R}_{\mathcal{L}}(x)$. Thus, by the witness indistinguishability of the WI system,

$$\mathsf{OUT}_{\mathsf{V}^*_\lambda} \langle \mathsf{P}(w), \mathsf{V}^*_\lambda \rangle (x)) \approx_{\mathsf{D}_\lambda, \lambda^{-\omega(1)}} \mathsf{S}(x, \mathsf{V}^*_\lambda, \mathsf{D}_\lambda, 1^{1/\varepsilon}) \ .$$

Case 2: $\Delta \leq \varepsilon$**.** Here we consider two sub-cases according to whether the simulator still obtains a ciphertext $\widehat{\mathsf{ct}}$ such that $\widetilde{\mathbf{CC}}(\widehat{\mathsf{ct}}) = 1$ or not. In case it does obtain such $\widehat{\mathsf{ct}}$, the analysis proceeds exactly as in case 1.

Henceforth, we assume that the simulator does not obtain such a witness, in which case, the simulator samples $\mathsf{ct}'' \leftarrow \mathsf{RSR.Enc}_{\mathsf{pk}'}(0; r')$ and uses r' as the witness for computing wi_2.

To complete the proof of validity we prove that in this case

$$\mathsf{RSR.Enc}_{\mathsf{pk}'}(0; r'), \mathsf{wi}_2(r') \approx_{\mathsf{D}_\lambda, \varepsilon + \lambda^{-\omega(1)}} \mathsf{RSR.Enc}_{\mathsf{pk}'}(1; r'), \mathsf{wi}_2(w) \enspace ,$$

where $\mathsf{wi}_2(r')$ and $\mathsf{wi}_2(w)$ denote the WI messages computed using the witnesses r' and w, respectively.

To prove this we consider several hybrids and prove that they are indistinguishable.

$\mathcal{H}_0$**:** In this hybrid, the prover message is generated as $\mathsf{RSR.Enc}_{\mathsf{pk}'}(1; r'), \mathsf{wi}_2(w)$ like in the real protocol.

$\mathcal{H}_1$**:** In this hybrid, the WI message wi_2 is generated using as as a witness the randomness r attesting that $\mathsf{cmt} = \mathsf{Com}(0; r)$ instead of using w.

This hybrid is indistinguishable from the previous one by the witness indistinguishability of the proof system.

$\mathcal{H}_2$**:** In this hybrid, the ciphertext $\mathsf{ct}'' \leftarrow \mathsf{RSR.Enc}_{\mathsf{pk}'}(0)$ is generated as an encryption of zero instead of one.

This hybrid is ε-indistinguishable from the previous one by our assumption that $\Delta \leq \varepsilon$.

$\mathcal{H}_3$**:** In this hybrid, the WI message wi_2 is generated using as as a witness the randomness r' attesting that $\mathsf{ct}'' = \mathsf{RSR.Enc}_{\mathsf{pk}'}(0; r')$ instead of using the randomness r for the commitment.

This hybrid is indistinguishable from the previous one by the witness indistinguishability of the proof system.

In this hybrid, the prover message is generated as $\mathsf{RSR.Enc}_{\mathsf{pk}'}(0; r'), \mathsf{wi}_2(r')$ like in the simulated protocol.

Overall, overall it follows that

$$\mathsf{OUT}_{\mathsf{V}^*_\lambda} \langle \mathsf{P}(w), \mathsf{V}^*_\lambda \rangle (x)) \approx_{\mathsf{D}_\lambda, \varepsilon + \lambda^{-\omega(1)}} \mathsf{S}(x, \mathsf{V}^*_\lambda, \mathsf{D}_\lambda, 1^{1/\varepsilon}) \enspace .$$

□

Remark 3.1 (On using relaxed random self-reducible encryption). We rely on relaxed RSR encryption by slightly tweaking the protocol. Specifically, in the protocol, the encryption ct'' will be under the alternative RSR encryption algorithm $\mathsf{RSR}.\widetilde{\mathsf{Enc}}$. The simulation and analysis remain the same.

4 Witness Hiding against Explainable Verifiers with Public Verification

In this section, relying on witness encryption, we give a variant of the protocol from the previous section that is also publicly-verifiable. The new protocol, however, is only WH and WZK, which is inherent (see introduction). We start by presenting the protocol and then proceed to analyze it.

Ingredients and notation:

- A 2-message publicly-verifiable WI argument for **NP** with delayed input. We denote its messages by $(\mathsf{wi}_1, \mathsf{wi}_2)$.

- A non-interactive perfectly-binding commitment scheme Com.

- A fully-homomorphic encryption scheme $(\mathsf{FHE.Enc}, \mathsf{FHE.Dec}, \mathsf{FHE.Eval})$.

- A compute-and-compare obfuscator $\mathcal{O}$.

- A witness encryption scheme $(\mathsf{WE.Enc}, \mathsf{WE.Dec})$.

We describe the protocol in Figure 2.

Protocol 2

Common Input: an instance $x \in \mathcal{L} \cap \{0,1\}^\lambda$, for security parameter λ.

P**'s auxiliary input:** a witness $w \in \mathcal{R}_\mathcal{L}(x)$.

1. V computes
   - wi_1, the first message of the WI argument,
   - $\mathsf{cmt} \leftarrow \mathsf{Com}(0; r)$, a commitment to zero, using randomness $r \leftarrow \{0,1\}^\lambda$,
   - $\mathsf{ct} \leftarrow \mathsf{FHE.Enc}_{\mathsf{sk}}(r)$, an encryption of the commitment randomness, under a randomly chosen secret key $\mathsf{sk} \leftarrow \{0,1\}^\lambda$.
   - $\widetilde{\mathbf{CC}} \leftarrow \mathcal{O}(\mathbf{CC}[\mathsf{FHE.Dec}_{\mathsf{sk}}, u])$, an obfuscation of the CC program given by the FHE decryption circuit and a random target $u \leftarrow \{0,1\}^\lambda$.
   - $\mathsf{ct}' \leftarrow \mathsf{WE.Enc}_x(u)$, a witness encryption of the target u.

   It sends $(\mathsf{wi}_1, \mathsf{cmt}, \mathsf{ct}, \widetilde{\mathbf{CC}}, \mathsf{ct}')$.

2. P computes wi_2, the second WI message for the statement $\Psi(x, \mathsf{cmt}, \widetilde{\mathbf{CC}})$ given by:

$$\begin{array}{lll} \exists w & : & (x,w) \in \mathcal{R}_\mathcal{L} & \bigvee \\ \exists r & : & \mathsf{cmt} = \mathsf{Com}(0; r) & \bigvee \\ \exists \widehat{\mathsf{ct}} & : & \widetilde{\mathbf{CC}}(\widehat{\mathsf{ct}}) = 1 \ , & \end{array}$$

   using the witness $w \in \mathcal{R}_\mathcal{L}(x)$. It sends wi_2.

3. V verifies the WI argument $(\mathsf{wi}_1, \mathsf{wi}_2)$ for the statement Ψ.

Figure 2: A publicly-verifiable 2-message WH argument $\langle \mathsf{P}, \mathsf{V} \rangle$ for **NP**.

Public verification. The verification of an argument in the above system amounts to applying the public verification of the WI argument, and involves no private randomness.

4.1 Analysis

We now analyze the protocol. We first show that it is sound, and then that it is WH against explainable verifiers.

Proposition 4.1. *Protocol 2 is sound.*

The soundness analysis is very similar to that of Protocol 1 (and in fact slightly simpler). We include it here for completeness (a reader who already went through the latter proof, may want to skip this one).

Proof. To prove soundness, we consider several hybrid protocols, and show that the probability that a malicious prover can cheat is preserved throughout the hybrids.

$\mathcal{H}_0$**:** This is the real protocol.

$\mathcal{H}_1$**:** In this hybrid, ct is an encryption of 0^λ instead of the target u.

Since $x \notin \mathcal{L}$, this hybrid is indistinguishable from the previous one by the semantic security of witness encryption scheme WE.

$\mathcal{H}_2$**:** In this hybrid, the obfuscation $\widetilde{\mathbf{CC}} \leftarrow \mathsf{Sim}(1^\lambda, 1^\ell)$ is simulated rather than an obfuscation of $\mathbf{CC}[\mathsf{FHE.Dec}_{\mathsf{sk}}, u]$, where ℓ is the size of the decryption circuit $\mathsf{FHE.Dec}_{\mathsf{sk}}$.

This hybrid is indistinguishable from the previous one by the CC simulation guarantee; indeed, in the previous hybrid, the target u is uniformly random and independent of $\mathsf{FHE.Dec}_{\mathsf{sk}}$, and the rest of the experiment.

$\mathcal{H}_3$**:** In this hybrid, ct is an encryption of 0^λ instead of the commitment randomness r.

This hybrid is indistinguishable from the previous one by the semantic security of the FHE scheme FHE.

$\mathcal{H}_4$**:** In this hybrid, cmt is a commitment to 1 instead of 0.

This hybrid is indistinguishable from the previous one by hiding of the commitment.

It is left to show that in $\mathcal{H}_4$, no malicious prover can convince the verifier to accept a false statement $x \notin \mathcal{L}$, except with negligible probability.

Observe that in this hybrid:

- $x \notin \mathcal{L}$.
- $\mathsf{cmt} \in \mathsf{Com}(1)$, and by perfect binding there does not exist r such that $\mathsf{cmt} = \mathsf{Com}(0; r)$.
- By the CC simulation guarantee, except with negligible probability, there does not exist $\widehat{\mathsf{ct}}$ such that $\widetilde{\mathbf{CC}}(\widehat{\mathsf{ct}}) = 1$.

Overall we deduce that the statement $\Psi(x, \mathsf{cmt}, \widetilde{\mathbf{CC}})$ is false. Soundness now follows by the soundness of the WI argument. □

Proposition 4.2. *Protocol 2 is witness hiding against explainable verifiers.*

Proof. We describe the witness-finding reduction R.

$\mathsf{R}(x, \mathsf{V}^*_\lambda, 1^{1/\varepsilon})$**:**

- Obtains $(\mathsf{wi}_1, \mathsf{cmt}, \mathsf{ct}, \widetilde{\mathbf{CC}}, \mathsf{ct}')$ from V^*_λ.
- Constructs the (*homomorphic simulation*) circuit $\mathsf{HS}(r)$ that given an input r:
  - Samples a second WI message wi_2 for the statement $\Psi(x, \mathsf{cmt}, \widetilde{\mathbf{CC}})$, using as the witness the randomness r attesting that $\mathsf{cmt} = \mathsf{Com}(0; r)$.
  - Feeds wi_2 to V^*_λ and obtains a candidate witness $\tilde{w}$.
  - Applies the witness decryptor
  $$\tilde{u} \leftarrow \mathsf{WE.Dec}_{\tilde{w}}(\mathsf{ct}') \enspace ,$$
  and outputs $\tilde{u}$.

  All randomness required by the above is hardwired to HS.
- Compute $\widehat{\mathsf{ct}} = \mathsf{FHE.Eval}(\mathsf{HS}, \mathsf{ct})$.
- If $\widetilde{\mathbf{CC}}(\widehat{\mathsf{ct}}) = 1$, repeat the following at most $1/\varepsilon$ times:

- Sample a second WI message wi_2 for the statement $\Psi(x, \mathsf{cmt}, \widetilde{\mathbf{CC}})$, using as the witness $\widehat{\mathsf{ct}}$ attesting that $\widetilde{\mathbf{CC}}(\widehat{\mathsf{ct}}) = 1$, feed it to V^*_λ.
- If V^*_λ outputs a witness $w \in \mathcal{R}_\mathcal{L}(x)$, output w.

- Otherwise, output $\bot$.

Reduction validity. The reduction clearly runs in polynomial time. We focus on proving its validity.

To prove validity, let $\mathsf{V}^* = \{\mathsf{V}^*_\lambda\}_\lambda$ be any explainable polynomial-size verifier. Fix any λ, and $x \in \mathcal{L} \cap \{0,1\}^\lambda$, and assume that V^*_λ outputs a witness $w \in \mathcal{R}_\mathcal{L}(x)$ with probability δ.

Let r be the randomness underlying the commitment $\mathsf{cmt} = \mathsf{Com}(0; r)$ given by V^*_λ. We argue that the circuit $\mathsf{HS}(r)$ outputs the target u underlying the CC obfuscation $\widetilde{\mathbf{CC}}$ with probability $\delta - \lambda^{-\omega(1)}$, over its own coins.

To see this, note that by witness indistinguishability, when HS gives V^*_λ the second WI message wi_2, V^*_λ outputs a witness w with probability $\delta - \lambda^{-\omega(1)}$. In this case, the witness decryption operation HS performs next, will indeed result in the target u.

By the correctness of FHE, the ciphertext $\widehat{\mathsf{ct}} = \mathsf{FHE.Eval}(\mathsf{HS}, \widehat{\mathsf{ct}})$ obtained by the reduction satisfies:

$$\mathsf{FHE.Dec}_{\mathsf{sk}}(\widehat{\mathsf{ct}}) = u \ ,$$

and thus by the one-sided correctness of the CC obfuscator $\mathcal{O}$,

$$\widetilde{\mathbf{CC}}(\widehat{\mathsf{ct}}) = 1 \ .$$

It follows that in this case, except with negligible probability $\lambda^{-\omega(1)}$, the reduction obtains a valid witness $\widehat{\mathsf{ct}}$ for the statement Ψ. By witness indistinguishability given the second WI message wi_2, using $\widehat{\mathsf{ct}}$ as the witness, V^*_λ outputs a witness with probability $\delta - \lambda^{-\omega(1)}$. Thus, by Markov's inequality, in this case, the reduction (which makes $1/\varepsilon$ attempts), obtains a witness with probability at least $1 - \frac{\varepsilon}{\delta} - \lambda^{-\omega(1)}$.

Overall, the reduction obtains a witness with probability at least

$$\left(\delta - \lambda^{-\omega(1)}\right) \cdot \left(1 - \frac{\varepsilon}{\delta} - \lambda^{-\omega(1)}\right) = \delta - \varepsilon - \lambda^{-\omega(1)} \ ,$$

as required. □

5 Privacy-Preserving Transformations from Explainable to Malicious Verifiers

In this section, we present three generic transformations that compile protocols that are private (according to some natural notion, such as ZK, WZK, WH, WI) against explainable verifiers into ones that satisfy the same privacy guarantee against malicious verifiers.

This includes the following:

- A 3-message transformation that applies for any language in **NP**, and preserves any notion of privacy.

- A 2-message transformation that applies for any language in **NP** $\cap$ **coNP**, and preserves any notion of privacy.

- A 2-message transformation that applies for any language in **NP**, but preserves only the notion of WH.

5.1 The 3-Message Transformation for NP.

For simplicity of exposition (and given the relevance to this paper), we describe the transformation for the case that the original protocol is also a 3-message one. The transformation easily extends to protocols with more messages, while preserving the round complexity.

Ingredients and notation:

- A 2-message WI argument for **NP** with delayed input. We denote its messages by $(\mathsf{wi}_1, \mathsf{wi}_2)$.
- A 3-message WI argument of knowledge for **NP**. We denote its messages by $(\mathsf{wik}_1, \mathsf{wik}_2, \mathsf{wik}_3)$.
- A non-interactive perfectly-binding commitment scheme Com.
- A 3-message argument system $\langle \mathsf{P}, \mathsf{V} \rangle$ for an **NP** language $\mathcal{L}$. We denote its messages by $(\mathsf{arg}_1, \mathsf{arg}_2, \mathsf{arg}_3)$.

We describe the protocol in Figure 3.

Protocol 3

Common Input: an instance $x \in \mathcal{L} \cap \{0,1\}^\lambda$, for security parameter λ.

$\bar{\mathsf{P}}$**'s auxiliary input:** a witness $w \in \mathcal{R}_{\mathcal{L}}(x)$.

1. $\bar{\mathsf{P}}$ computes
   - arg_1, the first message in the original protocol.
   - $\mathsf{cmt}_0, \mathsf{cmt}_1 \leftarrow \mathsf{Com}(0^{|w|})$, two independent commitments to zero strings.
   - wik_1, the first message of a WI argument of knowledge for the statement $\Psi(\mathsf{cmt}_0, \mathsf{cmt}_1)$ given by:
   $$\exists r_0, s_0 \ : \ \mathsf{cmt}_0 = \mathsf{Com}(s_0; r_0) \ \bigvee \ \exists r_1, s_1 \ : \ \mathsf{cmt}_1 = \mathsf{Com}(s_1; r_1) \ \ ,$$
   using (r_0, s_0) as the witness.
   - wi_1, the first message of a WI argument.

   It sends $(\mathsf{arg}_1, \mathsf{cmt}_0, \mathsf{cmt}_1, \mathsf{wik}_1, \mathsf{wi}_1)$.

2. $\bar{\mathsf{V}}$ computes
   - arg_2, the second message in the original protocol.
   - wik_2, the second message of the WI argument of knowledge for $\Psi(\mathsf{cmt}_0, \mathsf{cmt}_1)$.
   - wi_2, the second message of the WI argument for the statement $\Phi(x, \mathsf{cmt}_0, \mathsf{cmt}_1, \mathsf{arg}_1, \mathsf{arg}_2)$ given by:
   $$\begin{array}{lll} \exists r \ : \ \mathsf{arg}_2 = \mathsf{V}(x, \mathsf{arg}_1; r) & & \bigvee \\ \exists r_0, s_0 \ : & \begin{array}{c} \mathsf{cmt}_0 = \mathsf{Com}(s_0; r_0) \\ s_0 \notin \mathcal{R}_{\mathcal{L}}(x) \end{array} & \bigvee \\ \exists r_1, s_1 \ : & \begin{array}{c} \mathsf{cmt}_1 = \mathsf{Com}(s_1; r_1) \\ s_1 \notin \mathcal{R}_{\mathcal{L}}(x) \end{array} & . \end{array}$$

   It sends $(\mathsf{arg}_2, \mathsf{wik}_2, \mathsf{wi}_2)$.

3. $\bar{\mathsf{P}}$ verifies the WI argument $(\mathsf{wi}_1, \mathsf{wi}_2)$ for the statement Φ, and aborts if it does not accept.

   It then computes
   - arg_3, the third message in the original protocol.
   - wik_3, the third message of the WI argument of knowledge $\Psi(\mathsf{cmt}_0, \mathsf{cmt}_1)$.

   It sends $(\mathsf{arg}_3, \mathsf{wik}_3)$.

4. $\bar{\mathsf{V}}$ verifies
   - the WI argument $(\mathsf{wik}_1, \mathsf{wik}_2, \mathsf{wik}_3)$ for the statement Ψ,
   - the original argument $(\mathsf{arg}_1, \mathsf{arg}_2, \mathsf{arg}_3)$.

Figure 3: An argument $\langle \bar{\mathsf{P}}, \bar{\mathsf{V}} \rangle$ for **NP** against malicious verifiers.

Analysis. We now analyze the transformation.

Proposition 5.1. *Protocol 3 is sound.*

Proof. To prove soundness, we show how to transform any cheating prover $\bar{\mathsf{P}}^*$ against Protocol 3 into a cheating prover P^* against the original protocol.

Fix any polynomial-size prover $\bar{\mathsf{P}}^* = \left\{\bar{\mathsf{P}}^*_\lambda\right\}_\lambda$. We describe a new prover P^*, and show that for any $x \notin \mathcal{L}$, if $\bar{\mathsf{P}}^*$ convinces $\bar{\mathsf{V}}$ to accept with probability ε, the new prover convinces P^* convinces V to accept with probability $\varepsilon - \lambda^{-\omega(1)}$.

${\mathsf{P}^*}^{\bar{\mathsf{P}}^*}(x, 1^{1/\varepsilon})$**:**

- Constructs from $\bar{\mathsf{P}}^*$ a prover for the WI argument of knowledge $\mathsf{P}^*_{\mathsf{wik}}$ that works as follows:
  - Obtains $(\mathsf{arg}_1, \mathsf{cmt}_0, \mathsf{cmt}_1, \mathsf{wik}_1, \mathsf{wi}_1)$ from $\bar{\mathsf{P}}^*$, and sends wik_1 as the first message.
  - Given wik_2 from the WI verifier, it honestly emulates a second message arg_2 of V, and honestly emulates the WI argument second message wi_2. It feeds $(\mathsf{arg}_2, \mathsf{wik}_2, \mathsf{wi}_2)$ to $\bar{\mathsf{P}}^*$.
  - It obtains $(\mathsf{arg}_3, \mathsf{wik}_3$, and sends wik_3 as the third message.

  It now applies the knowledge extractor $\mathsf{E}^{\mathsf{P}^*_{\mathsf{wik}}}(x, 1^{1\varepsilon})$ to obtain a witness for the statement $\Psi(\mathsf{cmt}_0, \mathsf{cmt}_1)$.

- If the extractor fails to find a witness, P^* aborts. Otherwise, let (r, s) be the extracted witness; that is, an opening to cmt_0 or cmt_1.

- P^* now proceeds to its interaction with V.

- It sends arg_1 to V, and obtains arg_2.

- It emulates the argument of knowledge message wik_2 honestly, and computes the argument message wi_2 for $\Phi(x, \mathsf{cmt}_0, \mathsf{cmt}_1, \mathsf{arg}_1, \mathsf{arg}_2)$ using the witness (r, s) previously obtained.

- It then feeds $(\mathsf{arg}_2, \mathsf{wik}_2, \mathsf{wi}_2)$ to $\bar{\mathsf{P}}^*$, and obtains back $(\mathsf{arg}_3, \mathsf{wik}_3)$. It sends arg_3 to V.

Prover analysis. First, note that P^* runs in polynomial time (in its input length $\lambda + 1/\varepsilon$).

We now analyze the success probability. First, by the extraction guarantee, in the extraction phase, we onbtain a witness (r, s) with probability $1 - \lambda^{-\omega(1)}$. Note that since $x \notin \mathcal{L}$, it necessarily holds that $s \notin \mathcal{R}_{\mathcal{L}}(x)$, and accordingly (r, s) is also a valid witness for any statment $\Phi(x, \mathsf{cmt}_0, \mathsf{cmt}_1, \mathrm{arg}_1, \mathrm{arg}_2)$, regardless of arg_2.

Next, observe that the only difference between the view of $\bar{\mathsf{P}}^*$ in a real interaction with $\bar{\mathsf{V}}$ and its view as emulated by P^* is that in the first the WI argument $(\mathsf{wi}_1, \mathsf{wi}_2)$ is computed using the randomness of the honest verifier V as the witness, whereas in the second it is computed using the extracted witness (r, s).

By the witness indistinguishability of the argument, it follows that P^* convinces V with the same probability ε that $\bar{\mathsf{P}}^*$ convinces $\bar{\mathsf{V}}$, up to a negligible difference. □

Proposition 5.2. *There exists a PPT simulator S such that for any polynomial-size (malicious) verifier $\bar{\mathsf{V}}^* = \left\{\bar{\mathsf{V}}^*_\lambda\right\}_\lambda$ against Protocol 3:*

- $\mathsf{V}^* := \mathsf{S}^{\bar{\mathsf{V}}^*}$ *is an explainable verifier against the original protocol* $\langle\mathsf{P}, \mathsf{V}\rangle$.

- *For any polynomial-size distinguisher* $\mathsf{D} = \{\mathsf{D}_\lambda\}_\lambda$ *there exists a negligible* μ *such that for any* $\lambda \in \mathbb{N}$ *and any* $x \in \mathcal{L} \cap \{0,1\}^\lambda$,

$$\mathsf{OUT}_{\bar{\mathsf{V}}^*}\langle\mathsf{P}(w), \bar{\mathsf{V}}^*\rangle(x) \approx_{\mathsf{D}_\lambda, \mu} \mathsf{OUT}_{\mathsf{V}^*}\langle\mathsf{P}(w), \mathsf{V}^*\rangle(x) \enspace .$$

Proof. We describe the simulator S.

$\mathsf{S}^{\bar{\mathsf{V}}^*}(x)$**:**

- Obtains arg_1 from P.
- Emulates the first message of $\bar{\mathsf{P}}$ by computing:
  - $\mathsf{cmt}_0, \mathsf{cmt}_1 \leftarrow \mathsf{Com}(0^{|w|})$.
  - wik_1, the first message of a WI argument of knowledge for the statement $\Psi(\mathsf{cmt}_0, \mathsf{cmt}_1)$. For this it uses $(r_0, s_0 = 0^w)$ as the witness.
  - wi_1, the first message of a WI argument.

  Feeds the emulated message $(\mathsf{arg}_1, \mathsf{cmt}_0, \mathsf{cmt}_1, \mathsf{wik}_1, \mathsf{wi}_1)$ to the oracle $\bar{\mathsf{V}}^*$, and obtains $(\mathsf{arg}_2, \mathsf{wik}_k, \mathsf{wi}_2)$.
- Verifies the WI argument $(\mathsf{wi}_1, \mathsf{wi}_2)$ for the statement $\Phi(x, \mathsf{cmt}_0, \mathsf{cmt}_1, \mathsf{arg}_1, \mathsf{arg}_2)$.
- If the argument is invalid, it simulates an abort message from $\bar{\mathsf{P}}$, feeds it to $\bar{\mathsf{V}}^*$, obtains its output, and outputs the same (aborting the interaction with P).
- Otherwise, it sends arg_2 to P and obtains arg_3.
- Emulates the third message (of $\bar{\mathsf{P}}$) by computing wik_3, the third message of the WI argument of knowledge for the statement $\Psi(\mathsf{cmt}_0, \mathsf{cmt}_1)$.
- Feeds the emulated message $(\mathsf{arg}_3, \mathsf{wik}_3)$ to $\bar{\mathsf{V}}^*$, obtains its output, and outputs the same.

Simulator analysis. The above simulator clearly runs in polynomial time. Furthermore, by construction, the view of the emulated $\bar{\mathsf{V}}^*$ is distributed identically to its view in an interaction with $\bar{\mathsf{P}}$.

From hereon, we focus on proving that $\mathsf{V}^* = \mathsf{S}^{\bar{\mathsf{V}}^*}$ is explainable. For this, it suffices to show that with overwhelming probability, unless arg_2 is explainable (i.e., $\mathsf{arg}_2 = \mathsf{V}(x, \mathsf{arg}_1; r)$ for some r), the WI argument $(\mathsf{wi}_1, \mathsf{wi}_2)$ given by $\bar{\mathsf{V}}^*$ is rejected. To show this, we consider several hybrids and show that the probability that the above event occurs is preserved throughout the hybrids.

$\mathcal{H}_0$**:** This is the real protocol.

$\mathcal{H}_1$**:** In this hybrid, the commitment cmt_1 is a commitment to $w \in \mathcal{R}_{\mathcal{L}}(x)$ instead of $0^{|w|}$.

This hybrid is indistinguishable from the previous one by the hiding of the commitment.

$\mathcal{H}_2$**:** In this hybrid, the WI argument of knowledge $(\mathsf{wik}_1, \mathsf{wik}_2, \mathsf{wik}_3)$ is computed using $(r_1, s_1 = w)$ as the witness instead of $(r_0, s_0 =^{|w|})$.

This hybrid is indistinguishable from the previous one by the witness indistinguishability of the argument.

$\mathcal{H}_3$**:** In this hybrid, the commitment cmt_0 is a commitment to $w \in \mathcal{R}_{\mathcal{L}}(x)$ instead of $0^{|w|}$.

This hybrid is indistinguishable from the previous one by the hiding of the commitment.

It is left to show that in $\mathcal{H}_3$, except with negligible probability, $\bar{\mathsf{V}}^*$ fails to produce an accepting WI argument unless arg_2 is explainable.

In this hybrid both $\mathsf{cmt}_0, \mathsf{cmt}_1$ are commitments to a valid witness, and by the perfect binding of Com, they cannot be opened to a non-witness. Accordingly, $\Phi(x, \mathsf{cmt}_0, \mathsf{cmt}_1, \mathsf{arg}_1, \mathsf{arg}_2)$ is true only if $\mathsf{arg}_2 \in \mathsf{V}(x, \mathsf{arg}_1)$. By the soundness of the WI argument, it follows that except with negligible probability $(\mathsf{wi}_1, \mathsf{wi}_2)$ is rejected unless arg_2 is explainable. □

5.2 The 2-Message Transformation for NP ∩ coNP.

Here we only consider the case that the original protocol is also a 2-message one.

Ingredients and notation:

- A non-interactive WI proof for **NP**. We denote the proof by niwi.
- A 2-message argument system $\langle \mathsf{P}, \mathsf{V} \rangle$ for an **NP** ∩ **coNP** language $\mathcal{L}$. We denote its messages by $(\mathsf{arg}_1, \mathsf{arg}_2)$.

The protocol can be viewed as a simplified version of Protocol 3, where the verifier, rather than arguing that the prover's commitment is to a non-witness, argues directly that the instance x is a no-instance, which is indeed an **NP** statement since $\mathcal{L} \in$ **coNP**.

We describe the protocol in Figure 4.

Protocol 4

Common Input: an instance $x \in \mathcal{L} \cap \{0,1\}^\lambda$, for security parameter λ.

$\bar{\mathsf{P}}$**'s auxiliary input:** a witness $w \in \mathcal{R}_{\mathcal{L}}(x)$.

1. $\bar{\mathsf{V}}$ computes
   - arg_1, the first message in the original protocol.
   - niwi, a WI proof for the statement $\Phi(x, \mathsf{arg}_1)$ given by:

   $$\exists r \ : \ \mathsf{arg}_1 = \mathsf{V}(x; r) \ \bigvee \ \exists \bar{w} \ : \ \bar{w} \in \mathcal{R}_{\{0,1\}^* \setminus \mathcal{L}}(x) \ .$$

   It sends $(\mathsf{arg}_1, \mathsf{niwi})$.
2. $\bar{\mathsf{P}}$ verifies the NIWI proof niwi for the statement Φ, and aborts if it does not accept.

   It then computes arg_2, the second message in the original protocol, and sends it.
3. $\bar{\mathsf{V}}$ verifies the original argument $\mathsf{arg}_1, \mathsf{arg}_2$.

Figure 4: An argument $\langle \bar{\mathsf{P}}, \bar{\mathsf{V}} \rangle$ for **NP** ∩ **coNP** against malicious verifiers.

Analysis. We now analyze the transformation.

Proposition 5.3. *Protocol 4 is sound.*

Proof. To prove soundness, we show how to transform any cheating prover $\bar{\mathsf{P}}^*$ against Protocol 3 into a cheating prover P^* against the original protocol. The constructed P^* requires a witness for the fact that the instance is a no-instance. Such a witness can be obtained non-uniformly (the reduction is uniform, however, provided the witness).

Fix any polynomial-size prover $\bar{\mathsf{P}}^* = \{\bar{\mathsf{P}}^*_\lambda\}_\lambda$. We describe a new prover P^*, and show that for any $(x, \bar{w}) \in \mathcal{R}_{\{0,1\}^* \setminus \mathcal{L}}$, if $\bar{\mathsf{P}}^*$ convinces $\bar{\mathsf{V}}$ to accept with probability ε, the new prover convinces P^* convinces V to accept with probability $\varepsilon - \lambda^{-\omega(1)}$.

$\mathsf{P}^{*\bar{\mathsf{P}}^*}(x, \bar{w})$**:**

- Obtains arg_1 from V.
- Computes the NIWI proof niwi for $\Phi(x, \mathsf{arg}_1)$ using the witness $\bar{w}$.
- It then feeds $(\mathsf{arg}_1, \mathsf{niwi})$ to $\bar{\mathsf{P}}^*$, and obtains back arg_2, which it sends to V.

Prover analysis. P^* clearly runs in polynomial time.

We analyze the success probability. Observe that the only difference between the view of $\bar{\mathsf{P}}^*$ in a real interaction with $\bar{\mathsf{V}}$ and its view as emulated by P^* is that in the first the NIWI proof niwi is computed using the randomness of the honest verifier V as the witness, whereas in the second it is computed using $\bar{w}$. By the witness indistinguishability of the argument, it follows that P^* convinces V with the same probability ε that $\bar{\mathsf{P}}^*$ convinces $\bar{\mathsf{V}}$, up to a negligible difference. □

Proposition 5.4. *There exists a PPT simulator* S *such that for any polynomial-size (malicious) verifier* $\bar{\mathsf{V}}^* = \left\{\bar{\mathsf{V}}^*_\lambda\right\}_\lambda$ *against Protocol 4:*

- $\mathsf{V}^* := \mathsf{S}^{\bar{\mathsf{V}}^*}$ *is an explainable verifier against the original protocol* $\langle \mathsf{P}, \mathsf{V}\rangle$.
- *For any polynomial-size distinguisher* $\mathsf{D} = \{\mathsf{D}_\lambda\}_\lambda$ *there exists a negligible* μ *such that for any* $\lambda \in \mathbb{N}$ *and any* $x \in \mathcal{L} \cap \{0,1\}^\lambda$,

$$\mathsf{OUT}_{\bar{\mathsf{V}}^*}\langle \mathsf{P}(w), \bar{\mathsf{V}}^*\rangle(x) \approx_{\mathsf{D}_\lambda,\mu} \mathsf{OUT}_{\mathsf{V}^*}\langle \mathsf{P}(w), \mathsf{V}^*\rangle(x) \ .$$

Proof. We describe the simulator S.

$\mathsf{S}^{\bar{\mathsf{V}}^*}(x)$**:**

- Obtains $(\mathsf{arg}_1, \mathsf{niwi})$ from $\bar{\mathsf{V}}^*$.
- Verifies the NIWI proof niwi for the statement $\Phi(x, \mathsf{arg}_1)$.
- If the argument is invalid, it simulates an abort message from $\bar{\mathsf{P}}$, feeds it to $\bar{\mathsf{V}}^*$, obtains its output, and outputs the same (aborting the interaction with P).
- Otherwise, it sends arg_2 to P and obtains arg_2.
- Feeds arg_2 to $\bar{\mathsf{V}}^*$, obtains its output, and outputs the same.

Simulator analysis. The above simulator clearly runs in polynomial time. Furthermore, by construction, the view of the emulated $\bar{\mathsf{V}}^*$ is distributed identically to its view in an interaction with $\bar{\mathsf{P}}$.

We prove that $\mathsf{V}^* = \mathsf{S}^{\bar{\mathsf{V}}^*}$ is explainable. For this, note that since $x \in \mathcal{L}$, $\Phi(x, \mathsf{arg}_1)$ is true only if $\mathsf{arg}_2 \in \mathsf{V}(x)$. By the soundness of the NIWI proof, it follows that niwi is rejected, and S aborts, unless arg_1 is explainable. □

5.3 The 2-Message WH Transformation for NP.

Here we again only consider the case that the original protocol is also a 2-message one.

Ingredients and notation:

- A witness encryption scheme WE for $\mathcal{L}$.
- A 2-message WH argument system $\langle \mathsf{P}, \mathsf{V}\rangle$ for $\mathcal{L}$. We denote its messages by $(\mathsf{arg}_1, \mathsf{arg}_2)$.

The protocol can be viewed as a variant of Protocol 4, where the verifier, rather than using a NIWI to prove that the instance x is a no-instance (or that it behaves honestly), for which $\mathcal{L}$ has to be in **coNP**, gives a witness encryption of its coins under x. This can be seen as a proof of honest behavior that is simulatable if $x \notin \mathcal{L}$, and otherwise is sound. The downside of this system is that it requires a witness $w \in \mathcal{R}_{\mathcal{L}}(x)$ in order to verify.

We describe the protocol in Figure 5.

Protocol 5

Common Input: an instance $x \in \mathcal{L} \cap \{0,1\}^\lambda$, for security parameter λ.

$\bar{\mathsf{P}}$'s auxiliary input: a witness $w \in \mathcal{R}_{\mathcal{L}}(x)$.

1. $\bar{\mathsf{V}}$ computes
   - arg_1, the first message in the original protocol.
   - $\mathsf{ct} \leftarrow \mathsf{WE.Enc}_x(r)$, a witness encryption of the coins r used to generate arg_1.

   It sends $(\mathsf{arg}_1, \mathsf{ct})$.
2. $\bar{\mathsf{P}}$ decrypts $\tilde{r} \leftarrow \mathsf{WE.Dec}_w(\mathsf{ct})$, and verifies that $\mathsf{arg}_1 = \mathsf{V}(x; r)$. If this is not the case, it aborts.
3. It then computes arg_2, the second message in the original protocol, and sends it.
4. $\bar{\mathsf{V}}$ verifies the original argument $\mathsf{arg}_1, \mathsf{arg}_2$.

Figure 5: An argument $\langle \bar{\mathsf{P}}, \bar{\mathsf{V}} \rangle$ for **NP** against malicious verifiers.

Analysis. We now analyze the transformation.

Proposition 5.5. *Protocol 5 is sound.*

Proof. To prove soundness, we show how to transform any cheating prover $\bar{\mathsf{P}}^*$ against Protocol **??** into a cheating prover P^* against the original protocol.

Fix any polynomial-size prover $\bar{\mathsf{P}}^* = \left\{\bar{\mathsf{P}}^*_\lambda\right\}_\lambda$. We describe a new prover P^*, and show that for any $x \notin \mathcal{L}$, if $\bar{\mathsf{P}}^*$ convinces $\bar{\mathsf{V}}$ to accept with probability ε, the new prover convinces P^* convinces V to accept with probability $\varepsilon - \lambda^{-\omega(1)}$.

$\mathsf{P}^{*\bar{\mathsf{P}}^*}(x)$**:**

- Obtains arg_1 from V.
- Computes a witness encryption of zeros $\mathsf{ct} \leftarrow \mathsf{WE.Enc}_x(0^\lambda)$.
- It then feeds $(\mathsf{arg}_1, \mathsf{ct})$ to $\bar{\mathsf{P}}^*$, and obtains back arg_2, which it sends to V.

Prover analysis. P^* clearly runs in polynomial time.

We analyze the success probability. Observe that the only difference between the view of $\bar{\mathsf{P}}^*$ in a real interaction with $\bar{\mathsf{V}}$ and its view as emulated by P^* is that in the first ct is an encryption of the randomness r of the honest verifier V, whereas in the second it is an encryption of zeros. Since $x \notin \mathcal{L}$, it follows by the security of the witness encryption that P^* convinces V with the same probability ε that $\bar{\mathsf{P}}^*$ convinces $\bar{\mathsf{V}}$, upto a negligible difference. □

Proposition 5.6. *Protocol 5 is witness hiding.*

Proof. Let R be the witness be the witness-finding reduction of the original protocol $\langle \mathsf{P}, \mathsf{V} \rangle$, we describe the witness-finding reduction $\bar{\mathsf{R}}$ for the new protocol $\langle \bar{\mathsf{P}}, \bar{\mathsf{V}} \rangle$. In what follows let $\bar{\mathsf{V}}^* = \left\{\bar{\mathsf{V}}^*_\lambda\right\}_\lambda$ be a polynomial-size verifier.

$\bar{\mathsf{R}}(x, \bar{\mathsf{V}}^*_\lambda, 1^{1/\varepsilon})$**:**

- Constructs from $\bar{\mathsf{V}}^*_\lambda$ a verifier V^*_λ for the original protocol defined as follows:
  - Runs $\bar{\mathsf{V}}^*_\lambda(x)$ and and obtains $(\mathsf{arg}_1, \mathsf{ct})$.
  - Emulates an abort message from $\bar{\mathsf{P}}$, feeds it to $\bar{\mathsf{V}}^*_\lambda$, and tests whether it outputs a witness $w \in \mathcal{R}_\mathcal{L}(x)$.
  - If so, it discards $\bar{\mathsf{V}}^*$'s message and sends P an honestly generated message $\mathsf{arg}'_1 \leftarrow \mathsf{V}(x)$, obtains arg' from P, and output w.
  - If $\bar{\mathsf{V}}^*_\lambda$ did not output a witness, it continues emulating the interaction using $\bar{\mathsf{V}}^*$'s message: it sends P the arg_1, obtains arg_2, feeds it to $\bar{\mathsf{V}}^*_\lambda$, obtains its output, and outputs the same.
- Runs $\mathsf{R}(x, \mathsf{V}^*_\lambda, 1^{1/\varepsilon})$.

Reduction analysis. The above reduction clearly runs in polynomial time.

We now analyze its validity. We first argue that

$$\Pr\left[\mathsf{OUT}_{\bar{\mathsf{V}}^*_\lambda}\langle \bar{\mathsf{P}}(w), \bar{\mathsf{V}}^*_\lambda\rangle(x) \in \mathcal{R}_\mathcal{L}(x)\right] \leq \Pr\left[\mathsf{HOUT}_{\mathsf{V}^*_\lambda}\langle \mathsf{P}(w), \mathsf{V}^*_\lambda\rangle(x) \in \mathcal{R}_\mathcal{L}(x)\right] \quad ;$$

namely, the probability that $\bar{\mathsf{V}}^*$ outputs a witness equals that probability that the honest truncation of V^* outputs one.

Indeed, we consider the following cases:

- Assume $\bar{\mathsf{V}}^*$ behaves maliciously, but outputs a witness given an abort message. Then, by construction, V^* behaves in an explainable manner (in fact, honestly) and outputs a witness.
- Assume $\bar{\mathsf{V}}^*$ is explainable. Then V^* either finds a witness when emulating an abort, or perfectly emulates the view of $\bar{\mathsf{V}}^*$, and thus outputs a witness with the same probability.

To conclude the proof, recall that any WH against explainable verifiers is also weakly witness hiding according to Lemma 2.2, and thus

$$\begin{aligned}
&\Pr\left[\mathsf{OUT}_{\bar{\mathsf{V}}^*_\lambda}\langle \bar{\mathsf{P}}(w), \bar{\mathsf{V}}^*_\lambda\rangle(x) \in \mathcal{R}_\mathcal{L}(x)\right] \leq \\
&\Pr\left[\mathsf{HOUT}_{\mathsf{V}^*_\lambda}\langle \mathsf{P}(w), \mathsf{V}^*_\lambda\rangle(x) \in \mathcal{R}_\mathcal{L}(x)\right] \leq \quad \Pr\left[\mathsf{R}(x, \mathsf{V}^*_\lambda, 1^{1/\varepsilon}) \in \mathcal{R}_\mathcal{L}(x)\right] + \varepsilon(\lambda) + \lambda^{-\omega(1)} = \\
&\qquad \Pr\left[\bar{\mathsf{R}}(x, \bar{\mathsf{V}}^*_\lambda, 1^{1/\varepsilon}) \in \mathcal{R}_\mathcal{L}(x)\right] + \varepsilon(\lambda) + \lambda^{-\omega(1)} \quad ,
\end{aligned}$$

concluding the proof. □

Remark 5.1. The transformation given by Protocol 5 preserves public verifiability. Indeed, verification is the same as in the original protocol.

References

[Bar01] Boaz Barak. How to go beyond the black-box simulation barrier. In *42nd Annual Symposium on Foundations of Computer Science, FOCS 2001, 14-17 October 2001, Las Vegas, Nevada, USA*, pages 106–115, 2001.

[BBK$^+$16] Nir Bitansky, Zvika Brakerski, Yael Tauman Kalai, Omer Paneth, and Vinod Vaikuntanathan. 3-message zero knowledge against human ignorance. In *Theory of Cryptography - 14th International Conference, TCC 2016-B, Beijing, China, October 31 - November 3, 2016, Proceedings, Part I*, pages 57–83, 2016.

[BCC$^+$17] Nir Bitansky, Ran Canetti, Alessandro Chiesa, Shafi Goldwasser, Huijia Lin, Aviad Rubinstein, and Eran Tromer. The hunting of the SNARK. *J. Cryptology*, 30(4):989–1066, 2017.

[BCPR14] Nir Bitansky, Ran Canetti, Omer Paneth, and Alon Rosen. 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*, pages 505–514, 2014.

[BD18] Zvika Brakerski and Nico Döttling. Two-message statistical sender-private OT from LWE. *IACR Cryptology ePrint Archive*, 2018:530, 2018.

[BGI$^+$17] Saikrishna Badrinarayanan, Sanjam Garg, Yuval Ishai, Amit Sahai, and Akshay Wadia. Two-message witness indistinguishability and secure computation in the plain model from new assumptions. In *Advances in Cryptology - ASIACRYPT 2017 - 23rd International Conference on the Theory and Applications of Cryptology and Information Security, Hong Kong, China, December 3-7, 2017, Proceedings, Part III*, pages 275–303, 2017.

[BGJ$^+$13] Elette Boyle, Sanjam Garg, Abhishek Jain, Yael Tauman Kalai, and Amit Sahai. Secure computation against adaptive auxiliary information. In *Advances in Cryptology - CRYPTO 2013 - 33rd Annual Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2013. Proceedings, Part I*, pages 316–334, 2013.

[BJY97] Mihir Bellare, Markus Jakobsson, and Moti Yung. Round-optimal zero-knowledge arguments based on any one-way function. In *Advances in Cryptology - EUROCRYPT '97, International Conference on the Theory and Application of Cryptographic Techniques, Konstanz, Germany, May 11-15, 1997, Proceeding*, pages 280–305, 1997.

[BKP18] Nir Bitansky, Yael Tauman Kalai, and Omer Paneth. Multi-collision resistance: a paradigm for keyless hash functions. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2018, Los Angeles, CA, USA, June 25-29, 2018*, pages 671–684, 2018.

[BL18] Nir Bitansky and Huijia Lin. One-message zero knowledge and non-malleable commitments. In *Theory of Cryptography Conference, TCC 2018, Goa, India, November 11-14, 2018, Proceedings*, 2018.

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

[BM14] Christina Brzuska and Arno Mittelbach. Indistinguishability obfuscation versus multi-bit point obfuscation with auxiliary input. In *Advances in Cryptology - ASIACRYPT 2014 - 20th International Conference on the Theory and Application of Cryptology and Information*

Security, Kaoshiung, Taiwan, R.O.C., December 7-11, 2014, Proceedings, Part II, pages 142–161, 2014.

[BOV07] Boaz Barak, Shien Jin Ong, and Salil P. Vadhan. Derandomization in cryptography. *SIAM J. Comput.*, 37(2):380–400, 2007.

[BP04a] Boaz Barak and Rafael Pass. On the possibility of one-message weak zero-knowledge. In *Theory of Cryptography, First Theory of Cryptography Conference, TCC 2004, Cambridge, MA, USA, February 19-21, 2004, Proceedings*, pages 121–132, 2004.

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

[BP12] Nir Bitansky and Omer Paneth. Point obfuscation and 3-round zero-knowledge. In *Theory of Cryptography - 9th Theory of Cryptography Conference, TCC 2012, Taormina, Sicily, Italy, March 19-21, 2012. Proceedings*, pages 190–208, 2012.

[BP15a] Nir Bitansky and Omer Paneth. On non-black-box simulation and the impossibility of approximate obfuscation. *SIAM J. Comput.*, 44(5):1325–1383, 2015.

[BP15b] Nir Bitansky and Omer Paneth. Zaps and non-interactive witness indistinguishability from indistinguishability obfuscation. In *Theory of Cryptography - 12th Theory of Cryptography Conference, TCC 2015, Warsaw, Poland, March 23-25, 2015, Proceedings, Part II*, pages 401–427, 2015.

[BP15c] Elette Boyle and Rafael Pass. Limits of extractability assumptions with distributional auxiliary input. In *Advances in Cryptology - ASIACRYPT 2015 - 21st International Conference on the Theory and Application of Cryptology and Information Security, Auckland, New Zealand, November 29 - December 3, 2015, Proceedings, Part II*, pages 236–261, 2015.

[BST16] Mihir Bellare, Igors Stepanovs, and Stefano Tessaro. Contention in cryptoland: Obfuscation, leakage and UCE. In *Theory of Cryptography - 13th International Conference, TCC 2016-A, Tel Aviv, Israel, January 10-13, 2016, Proceedings, Part II*, pages 542–564, 2016.

[BV14] Zvika Brakerski and Vinod Vaikuntanathan. Efficient fully homomorphic encryption from (standard) LWE. *SIAM J. Comput.*, 43(2):831–871, 2014.

[CLP13] Kai-Min Chung, Huijia Lin, and Rafael Pass. Constant-round concurrent zero knowledge from p-certificates. In *54th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2013, 26-29 October, 2013, Berkeley, CA, USA*, pages 50–59, 2013.

[CLP15] Kai-Min Chung, Edward Lui, and Rafael Pass. From weak to strong zero-knowledge and applications. In *Theory of Cryptography - 12th Theory of Cryptography Conference, TCC 2015, Warsaw, Poland, March 23-25, 2015, Proceedings, Part I*, pages 66–92, 2015.

[CPS16] Kai-Min Chung, Rafael Pass, and Karn Seth. Non-black-box simulation from one-way functions and applications to resettable security. *SIAM J. Comput.*, 45(2):415–458, 2016.

[CVW18] Yilei Chen, Vinod Vaikuntanathan, and Hoeteck Wee. GGH15 beyond permutation branching programs: Proofs, attacks, and candidates. In *Advances in Cryptology - CRYPTO 2018 - 38th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2018, Proceedings, Part II*, pages 577–607, 2018.

[DGS09] Yi Deng, Vipul Goyal, and Amit Sahai. Resolving the simultaneous resettability conjecture and a new non-black-box simulation strategy. In *50th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2009, October 25-27, 2009, Atlanta, Georgia, USA*, pages 251–260, 2009.

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

[DNRS03] Cynthia Dwork, Moni Naor, Omer Reingold, and Larry J. Stockmeyer. Magic functions. *J. ACM*, 50(6):852–921, 2003.

[FGJ18] Nils Fleischhacker, Vipul Goyal, and Abhishek Jain. On the existence of three round zero-knowledge proofs. In *Advances in Cryptology - EUROCRYPT 2018 - 37th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tel Aviv, Israel, April 29 - May 3, 2018 Proceedings, Part III*, pages 3–33, 2018.

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

[FS90] Uriel Feige and Adi Shamir. Witness indistinguishable and witness hiding protocols. In *Proceedings of the 22nd Annual ACM Symposium on Theory of Computing, May 13-17, 1990, Baltimore, Maryland, USA*, pages 416–426, 1990.

[Gam85] Taher El Gamal. A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Trans. Information Theory*, 31(4):469–472, 1985.

[Gen09a] Craig Gentry. *A fully homomorphic encryption scheme*. PhD thesis, Stanford University, 2009. crypto.stanford.edu/craig.

[Gen09b] Craig Gentry. Fully homomorphic encryption using ideal lattices. In *Proceedings of the 41st Annual ACM Symposium on Theory of Computing, STOC 2009, Bethesda, MD, USA, May 31 - June 2, 2009*, pages 169–178, 2009.

[GGH$^+$16] Sanjam Garg, Craig Gentry, Shai Halevi, Mariana Raykova, Amit Sahai, and Brent Waters. Candidate indistinguishability obfuscation and functional encryption for all circuits. *SIAM J. Comput.*, 45(3):882–929, 2016.

[GGSW13] Sanjam Garg, Craig Gentry, Amit Sahai, and Brent Waters. Witness encryption and its applications. In *Symposium on Theory of Computing Conference, STOC'13, Palo Alto, CA, USA, June 1-4, 2013*, pages 467–476, 2013.

[GHKW17] Rishab Goyal, Susan Hohenberger, Venkata Koppula, and Brent Waters. A generic approach to constructing and proving verifiable random functions. In *Theory of Cryptography - 15th International Conference, TCC 2017, Baltimore, MD, USA, November 12-15, 2017, Proceedings, Part II*, pages 537–566, 2017.

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

[GKW17] Rishab Goyal, Venkata Koppula, and Brent Waters. Lockable obfuscation. In *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017, Berkeley, CA, USA, October 15-17, 2017*, pages 612–621, 2017.

[GM84] Shafi Goldwasser and Silvio Micali. Probabilistic encryption. *J. Comput. Syst. Sci.*, 28(2):270–299, 1984.

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

[GMW91] Oded Goldreich, Silvio Micali, and Avi Wigderson. 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] Oded Goldreich and Yair Oren. Definitions and properties of zero-knowledge proof systems. *J. Cryptology*, 7(1):1–32, 1994.

[GOS12] Jens Groth, Rafail Ostrovsky, and Amit Sahai. New techniques for noninteractive zero-knowledge. *J. ACM*, 59(3):11:1–11:35, 2012.

[Goy13] Vipul Goyal. Non-black-box simulation in the fully concurrent setting. In *Symposium on Theory of Computing Conference, STOC'13, Palo Alto, CA, USA, June 1-4, 2013*, pages 221–230, 2013.

[HIK$^+$11] Iftach Haitner, Yuval Ishai, Eyal Kushilevitz, Yehuda Lindell, and Erez Petrank. Black-box constructions of protocols for secure computation. *SIAM J. Comput.*, 40(2):225–266, 2011.

[HRS09] Iftach Haitner, Alon Rosen, and Ronen Shaltiel. On the (im)possibility of arthur-merlin witness hiding protocols. In *Theory of Cryptography, 6th Theory of Cryptography Conference, TCC 2009, San Francisco, CA, USA, March 15-17, 2009. Proceedings*, pages 220–237, 2009.

[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.

[HW15] Pavel Hubáček and Daniel Wichs. On the communication complexity of secure function evaluation with long output. In *Proceedings of the 2015 Conference on Innovations in Theoretical Computer Science, ITCS 2015, Rehovot, Israel, January 11-13, 2015*, pages 163–172, 2015.

[JKKR17] Abhishek Jain, Yael Tauman Kalai, Dakshita Khurana, and Ron Rothblum. Distinguisher-dependent simulation in two rounds and its applications. In *Advances in Cryptology - CRYPTO 2017 - 37th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 20-24, 2017, Proceedings, Part II*, pages 158–189, 2017.

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

[OPP14] Rafail Ostrovsky, Anat Paskin-Cherniavsky, and Beni Paskin-Cherniavsky. Maliciously circuit-private FHE. In *Advances in Cryptology - CRYPTO 2014 - 34th Annual Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2014, Proceedings, Part I*, pages 536–553, 2014.

[Pai99] Pascal Paillier. Public-key cryptosystems based on composite degree residuosity classes. In *Advances in Cryptology - EUROCRYPT '99, International Conference on the Theory and Application of Cryptographic Techniques, Prague, Czech Republic, May 2-6, 1999, Proceeding*, pages 223–238, 1999.

[Pas03] Rafael Pass. Simulation in quasi-polynomial time, and its application to protocol composition. In *Advances in Cryptology - EUROCRYPT 2003, International Conference on the Theory and Applications of Cryptographic Techniques, Warsaw, Poland, May 4-8, 2003, Proceedings*, pages 160–176, 2003.

[Reg09] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. *J. ACM*, 56(6):34:1–34:40, 2009.

[WZ17] Daniel Wichs and Giorgos Zirdelis. Obfuscating compute-and-compare programs under LWE. In *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017, Berkeley, CA, USA, October 15-17, 2017*, pages 600–611, 2017.