Weak Zero-Knowledge Beyond the Black-Box Barrier

Nir Bitansky* Dakshita Khurana† Omer Paneth‡

July 31, 2019

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: none 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 two messages, assuming quasipolynomially-secure fully-homomorphic encryption and other standard primitives (known from quasipolynomial hardness of Learning with Errors), as well as subexponentially-secure one-way functions.
- Weak zero-knowledge for **NP** in three messages under standard polynomial assumptions (following for example from fully-homomorphic encryption and factoring).

We also give, under polynomial assumptions, a two-message witness-hiding protocol for any language $\mathcal{L} \in \mathbf{NP}$ that has a witness encryption scheme. 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 classic 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, by Len Blavatnik and the Blavatnik Family foundation, and an ISF grant 18/484.

†Microsoft Research, New England, dakshkhurana@gmail.com

‡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 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 witness indistinguishability is natural and often useful as a building block in applications, it 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 the order of quantifiers. Full-fledged zero-knowledge requires that for every verifier there exists a $\underline{\text{simulator}}$ that generates a view, indistinguishable from the verifier's view in a real interaction, for every $\underline{\text{distinguisher}}$. In contrast, weak zero-knowledge requires that for every verifier and $\underline{\text{distinguisher}}$, there exists a $\underline{\text{simulator}}$ that fools this specific distinguisher. We also allow the simulator to depend on the desired distinguishing gap.[2]

  Weak zero-knowledge hides any 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, constructions with less than four messages are known only based on non-standard assumptions, which by now are considered implausible

[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].

[BP12, BM14, BST16], for restricted classes of adversarial verifiers [BCPR14, JKKR17], or for restricted classes of languages [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 a non-standard assumption).

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.

Our main result is a two-message weak zero-knowledge argument for **NP**.

Theorem 1.1 (informal)**.** *There exists a two-message weak zero-knowledge argument for **NP** assuming: subexponentially-secure one-way functions and quasi-polynomially-secure fully-homomorphic encryption, random-self-reducible encryption, two-message witness-indistinguishable arguments and oblivious transfer, non-interactive commitments, and compute-and-compare obfuscation.*

All of the above primitives (but subexponentially-secure one-way functions) are known under quasipolynomial hardness of LWE, with the exception of fully-homomorphic encryption that also requires a circular security assumption [Gen09b, BV14, BGI$^+$17, GHKW17, GKW17, JKKR17, WZ17, BD18]. We can also replace compute-and-compare obfuscation with fully-homomorphic encryption scheme with some additional natural properties that are satisfied by known constructions (see the technical overview for more details).

It is interesting to note that the result gives an example of a *natural* weak-zero-knowledge protocol that is provably not zero knowledge. Previously, a contrived separation was known assuming exponentially-hard injective one-way functions [CLP15].

Our second result is a three-message protocol that is based only on polynomial hardness assumptions.

Theorem 1.2 (informal)**.** *Assuming polynomial hardness of the primitives in Theorem 1.1, as well as dense commitments, there exists a three-message weak zero-knowledge argument for **NP**.*

The polynomially-hard version of the required primitives (including dense commitments) can be based on (polynomially-hard) fully-homomorphic encryption, LWE, and either Factoring or standard Bilinear-Group assumptions.

Our third result, also from polynomial hardness assumptions, is a two-message witness-hiding protocol for any language $\mathcal{L} \in$ **NP** that has a *witness encryption scheme* [GGSW13].

Theorem 1.3 (informal)**.** *There exists a two-message witness-hiding argument for any language $\mathcal{L} \in$ **NP** under the same (polynomial) assumptions as in Theorem 1.2 and witness encryption for $\mathcal{L}$.*

For the time being, witness encryption for all of **NP** is only known based on indistinguishability obfuscation, or based on non-standard assumptions on multilinear maps [GGH$^+$16, CVW18]. Witness encryption for several non-trivial languages follows from results on hash proofs systems [CS02].

The protocol we obtain is *publicly verifiable*, meaning that the proof can be verified given the transcript alone, without 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 weak-zero-knowledge argument for **NP** against a new class of verifiers that we call *explainable*. Such verifiers may not follow the honest verifier strategy, but they do 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 that of *semi-malicious* and *defensible* adversaries [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 (respectively, 1.2), there exists a two-message (respectively, three-message) weak zero-knowledge argument for **NP** against explainable verifiers.*

We then give general compilers to boost explainable security to malicious security. These compilers may be of independent interest. For instance, they imply that to obtain full-fledged zero knowledge in three messages, it suffices to consider explainable verifiers.

1.2 Technical Overview

We now give an overview of our techniques. We focus on two-message protocols against explainable verifiers, which is the technical core behind our results. We also explain how to avoid super-polynomial hardness assumptions, at the account of adding one message to this protocol. 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 consider 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 for x with noticeable probability.

Our protocol follows a classic 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. (Throughout 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 x, τ should be computationally indistinguishable from a false statement. Thus, by the soundness of the witness-indistinguishable argument, 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 realizing the above paradigm is to extract 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, 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 — under the encryption ct, it uses 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 when x is a true statement; 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. Note that we crucially rely on the fact that a fully homomorphic encryption scheme satisfies *compactness* in the sense that the size of the encrypted witness (and hence the trapdoor τ_h) is fixed and independent of the size of the homomorphic computation performed to obtain this encryption. Indeed, this computation grows with the size of the malicious verifier, which is not apriori bounded.

One difficulty in realizing the above strategy is to prove the homomorphic trapdoor statement τ_h; that is, to prove 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 a 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 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 polynomial in $|\mathsf{D}|/\varepsilon$. 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 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. The statements τ, τ' also fix 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 one $\mathsf{ct_P} = \mathsf{RSR.Enc_{pk}}(1)$. The witness-indistinguishable argument attests that either one of the statements x, τ, τ_h hold, or $\mathsf{ct_P}$ is an encryption of zero (and not one). 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 one.

Soundness. Relying on the security of both encryption schemes, we would like to argue that 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 zero or the verifier rejects. Soundness then follows from that of the witness-indistinguishable argument.

One subtlety in the above argument is that the verifier's decision depends on the secret key of the the random-self-reducible encryption, and thus changing the encryption of ρ' to garbage may affect the bit underlying the prover's encryption $\mathsf{ct_P}$ and accordingly also the verifier's decision bit. To get around this, we require that the prover exhibits that it "knows" the contents of the encryption $\mathsf{ct_P}$ (and therefore does not maul the encryption of ρ'). In this case, we can argue that the verifier continues to accept, even if we change the encryption of ρ'. To facilitate such a proof of knowledge in two messages we resort to complexity leveraging, which is the cause of reliance on super-polynomial assumptions in our two-message protocol. In the three-message setting, we rely instead on extractable commitments based on polynomial hardness assumptions.

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 polynomial in $|\mathsf{D}|/\varepsilon$.

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 realizing this strategy is to extract 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 one $\mathsf{ct_P} = \mathsf{RSR.Enc_{pk}}(1)$. However, differently from the real proof distribution, the witness-indistinguishable argument is computed using the homomorphic trapdoor witness ρ_h.

- The public distribution Π_p consists of an encryption of zero $\mathsf{ct_P} = \mathsf{RSR.Enc_{pk}}(0)$, 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 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, D' has roughly the same advantage as 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 homomorphically applying the first step — under the encryption ct sent by the verifier, 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 we follow in the body of the paper) is based on obfuscation for *compute and compare programs*. A compute and compare 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].[3]

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 and 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 can obtain a fully-homomorphic encryption of ρ' and use it 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.

Another approach, without compute-and-compare obfuscation. We now describe an alternative approach for proving the trapdoor statement, which does not rely on compute-and-compare obfuscation, but requires the homomorphic encryption to satisfy additional properties.

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

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

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 ρ_h. 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]. (Also, a similar malleability problem as described before also occurs here and is dealt with using a proof of knowledge.)

Another issue is that the simulator cannot tell whether the encryption ρ_h it obtained is indeed an encryption of a valid trapdoor witness ρ' or not (in which case, it should deduce that D cannot tell Π from Π_p and use Π_p to simulate). To deal with this, we again use the distinguisher D' to decrypt ct', except that we do so in the clear. Since now we do not have the trapdoor witness ρ, we use ρ_h instead. By witness indistinguishability, if we fail to decrypt ct' and obtain the witness ρ' in the clear, we can deduce that D cannot tell Π from Π_p.

1.2.1 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 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 protocols secure against explainable verifiers and turn them into protocols secure against malicious verifiers. We provide three different compilers for different settings. We now explain the main ideas behind each of these compilers.

A two-message compiler based on super-polynomial assumptions. Our first compiler is based on two-message conditional disclosure of secrets schemes for **NP**, which is known under standard assumptions [AIR01, BP12, AJ17]. In such a scheme, the verifier first sends an instance x' for some **NP** language $\mathcal{L}'$ together with an encrypted witness. The prover responds with an encryption of a message, which the honest verifier can then decrypt. In contrast, if a cheating verifier sends $x' \notin \mathcal{L}'$, the prover's message is completely hidden.

The compiler works as follows. The parties emulate the original two-message protocol for explainable verifiers. The verifier also sends the first message of a conditional disclosure protocol for the statement x' asserting that its message is explainable (namely, it is in the support of the honest verifier's messages). The prover then responds with an encryption, relative to x', of its second message in the underlying protocol. The verifier decrypts and verifies the underlying protocol.

Intuitively, if the verifier does not behave in an explainable manner, the statement x' is false, the prover's message is hidden, and the verifier learns nothing. If the verifier is explainable, then the weak zero-knowledge guarantee of the underlying protocol kicks in. To argue soundness, we would like to give a reduction that can efficiently extract the prover's encrypted message, without using the verifier's randomness. To this end, the prover provides a proof-of-knowledge of its message. To enable such a proof of knowledge in only two messages, we again rely on complexity leveraging.

A three-message compiler based on polynomial assumptions. A natural approach toward a three-message compiler based on polynomial assumptions is to augment the previous compiler with a three-message proof of knowledge (rather than a two-message one based on complexity leveraging). However, we do not know how to prove that this approach works. In a nutshell, the issue is that the explainability of the verifier's message may now depend on the first prover message, and cannot be efficiently tested. Instead, we take a different approach inspired by [BP15b].

To understand the basic idea behind the compiler, imagine first that the language $\mathcal{L}$ is in $\mathbf{NP} \cap \mathbf{coNP}$. The compiler works as follows. Given the statement x, the verifier provides, together with its message, a witness-indistinguishable argument that either $x \notin \mathcal{L}$ or that its message is explainable; 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 compiler preserves the privacy guarantee of the original protocol. By the soundness of the witness-indistinguishable argument, 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. Given this witness as a non-uniform advice, the reduction can turn any cheating prover against the compiled scheme into a cheating prover against the original scheme. By witness indistinguishability, the reduction can use the witness for $x \notin \mathcal{L}$ to compute the witness-indistinguishable arguments without compromising the verifier's randomness.

To extend the above to all of **NP**, we use the first prover message to map the statement x into a related **coNP** statement. For this, we rely on perfectly binding *dense commitments* 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 proceeds to prove 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).

To argue soundness when $x \notin \mathcal{L}$, note that the dense commitment can necessarily be opened to some string, which is not a witness. The reduction then proceeds as in the previous protocol. Weak zero-knowledge follows by a simple extension of the previous argument. If the commitment in the first message indeed contains a valid witness, then by the binding of the commitment and the soundness of the verifier's proofs, the verifier fails to prove that the commitment is 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 can generate alone.

We note that this compiler actually preserves all natural security notions (like, zero-knowledge, weak zero-knowledge, or witness hiding).

Two-message witness-hiding compilers. We provide another two-message compiler for witness hiding protocol that is based on polynomial assumptions and is also publicly verifiable. The compiler works for any language $\mathcal{L} \in$ **NP** provided a witness encryption scheme for $\mathcal{L}$. moved this here: In a witness encryption scheme for $\mathcal{L}$, 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 compiler is as follows. Given the statement x, the verifier sends, together with its message, an encryption of its randomness under witness encryption, using the instance x as the public-key. The honest prover, holding 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, such a verifier does not obtain any information. However, this intuition is misleading — a malicious verifier may generate messages without knowing whether they are explainable, and use the prover's abort decision to learn this bit of information.[4] Nonetheless, the compiler does preserve witness hiding — the witness-finding reduction can simply guess if the verifier's message is explainable and simulate the prover's message accordingly. This only decreases its success probability by a factor of two.

1.3 More on Related Work

We next address related work in more detail.

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-

[4] In fact, for some instantiations of the witness encryption, this protocol reveals an arbitrary bit of the witness [BP15b].

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 on statements with at least two "independent" witnesses. In [BP12], three-message weak zero-knowledge and witness-hiding protocols are constructed based on non-standard assumptions, which by now are considered implausible [BM14, BST16]. Specifically, these constructions are based on the notion of recognizable auxiliary-input point obfuscation that was shown in [BST16] to be impossible assuming virtual-grey-box obfuscation exists.

Upgrading weak zero-knowledge. The work of [CLP15] considers two relaxed notions of zero knowledge and proves that they are equivalent to their weak variants (where the simulator can depend on the distinguisher). The first notion they consider is distributional zero-knowledge where instances are sampled from some known distribution (and the simulator can depend on this distribution). The second notion is zero knowledge against uniform distinguishers (these distinguishers can still get an auxiliary input, but the same input is given to the simulator). In both settings, the equivalence is shown for (t, ϵ)-zero-knowledge, where the distinguisher's running time and distinguishing gap are bounded by t and ϵ, respectively, and the simulator's running time can depend on t and $1/\epsilon$.

Combining our weak-zero-knowledge protocols with the equivalence theorems of [CLP15] yields distributional (t, ϵ)-zero-knowledge and (t, ϵ)-zero-knowledge against uniform distinguishers. Note that the two-message lower bound of [GO94] does not apply for these notions.

Distributional security against non-adaptive verifiers. The work of [JKKR17] constructs *distributional* weak-zero-knowledge and witness-hiding protocols for a restricted class of *non-adaptive verifiers* who choose their messages obliviously of the proven statement. They give protocols in three messages under standard assumptions, and in two messages under standard, but super-polynomial, assumptions. Their simulators and (witness-finding) reductions access the verifier as a black box.

Bounded description adversaries. Another type of relaxation considered in the literature is restricting the (adversarial) verifier or prover to a-priori bounded description (and arbitrary polynomial running time). Here (full-fledged) zero-knowledge can be constructed in two messages against bounded-description verifiers under standard, but super-polynomial, assumptions [BCPR14], and in three messages against bounded-description provers assuming also keyless hash functions that are collision-resistant against bounded-description adversaries [BBK$^+$16].

Super-polynomial simulation. Zero knowledge with simulators that run in super-polynomial time 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.

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 studied extensively. Four-message proofs are impossible to achieve via black-box simulation, except for languages in $\mathbf{NP} \cap \mathbf{coMA}$ [Kat12]. Four message proofs with non-black-box simulation are only known assuming multi-collision-resistance keyless hash functions [BKP18]. Recent evidence [FGJ18] suggests that, differently from zero-knowledge arguments, zero-knowledge proofs may be impossible to achieve in three messages (even with non-black-box simulation).

Honest-verifier zero knowledge. For languages in the class **SZK**, there exist two-message proofs with an inefficient prover (known as Arthur-Merlin proofs) that are zero knowledge only against honest verifiers [AH91, SV97]. Two-message honest-verifier zero-knowledge is implied by any NIZK with a common *random* string (such a NIZK exists under computational assumptions).

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 also consider probabilistic circuits that may toss random coins.
- 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.
- We also consider quasipolynomial-size adversaries, which are defined analogously to polynomial-size adversaries, but are of size $2^{(\log\lambda)^{O(1)}}$ instead of $\lambda^{O(1)}$. By default we define the security of primitives against polynomial-size adversaries. Security against quasipolynomial-size adversaries is always defined analogously.
- 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
$$|\Pr[\mathsf{D}(X) = 1] - \Pr[\mathsf{D}(Y) = 1]| \leq \mu.$$

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 $\lambda \in \mathbb{N}, x \in \mathcal{L} \cap \{0,1\}^\lambda$, $w \in \mathcal{R}_\mathcal{L}(x)$:*
$$\Pr\left[\mathsf{OUT}_\mathsf{V}\langle \mathsf{P}(w), \mathsf{V}\rangle(x) = 1\right] = 1 \ .$$
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 $x \in \{0,1\}^\lambda \setminus \mathcal{L}$,*
$$\Pr\left[\mathsf{OUT}_\mathsf{V}\langle \mathsf{P}^*_\lambda, \mathsf{V}\rangle(x) = 1\right] \leq \mu(\lambda) \ .$$

   The argument is **sound against quasipolynomial provers**, *if the above holds also for quasipolynomial-size* P^*.

Remark 2.1 (Public verification)*.* We say that the protocol is *publicly verifiable* if the verifier's decision bit can be computed from the protocol's messages (without verifier private state).

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

2.1.1 Weak Zero-Knowledge

Definition 2.2 (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 $\mathsf{V}^* = \{\mathsf{V}^*_\lambda\}_\lambda$, distinguisher $\mathsf{D} = \{\mathsf{D}_\lambda\}_\lambda$, and noticeable $\varepsilon(\lambda)$, there exists a negligible $\mu(\lambda)$, such that for any $\lambda \in \mathbb{N}$, $x \in \mathcal{L} \cap \{0,1\}^\lambda$, and $w \in \mathcal{R}_\mathcal{L}(x)$,*

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

Remark 2.4 (w.l.o.g). When convenient we assume w.l.o.g that V^* is deterministic. This is as in the case of (standard) zero knowledge with a universal simulator [GO94]. We do not assume that the distinguisher is deterministic. This is in contrast to standard zero-knowledge where we often fix the best random coins for the distinguisher. In weak zero-knowledge this is not possible, as the simulated distribution can depend on the distinguisher.

Also, when convenient we assume w.l.o.g that verifiers always output their entire view consisting of the prover message and, if they are probabilistic, their randomness.

2.1.2 Witness Hiding

Definition 2.3 (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 $x \in \mathcal{L} \cap \{0,1\}^\lambda$,*

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

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 Explainable Verifiers

Roughly speaking, explainable verifiers are verifiers whose messages are (almost) always in the support of the honest verifier's messages, regardless of how the prover's messages are generated. We also allow such verifiers to abort at any stage.

Definition 2.4 (Explainable transcript)**.** *Let $\langle \mathsf{P}, \mathsf{V}\rangle$ be a protocol, P^* be an arbitrary prover, and V^* an arbitrary verifier. We say that a transcript T of an execution $\langle \mathsf{P}^*, \mathsf{V}^*\rangle(x)$ is explainable if there exists honest verifier coins r such that T is consistent with the transcript of an execution $\langle \mathsf{P}^*, \mathsf{V}_r\rangle$ until the point in T that V^* aborts. (Here V_r is the honest verifier using coins r).*

Definition 2.5 (Explainable verifier)**.** *Let $\langle \mathsf{P}, \mathsf{V}\rangle$ be a protocol. A (possibly probabilistic) verifier $\mathsf{V}^* = \{\mathsf{V}^*_\lambda\}_\lambda$ is explainable if there exists a negligible $\mu(\lambda)$ such that for any prover P^*, $\lambda \in \mathbb{N}$, and $x \in \{0,1\}^\lambda$,*

$$\Pr_{\mathsf{V}^*_\lambda}\left[T \text{ is explainable} \mid T \leftarrow \langle \mathsf{P}^*, \mathsf{V}^*_\lambda\rangle(x)\right] \geq 1 - \mu(\lambda) \ .$$

In the two-message setting, We will also consider a simpler notion of *always explainable verifiers*, which are deterministic verifiers that always output a message in the support of the honest verifier.

Definition 2.6 (Always-explainable verifier)**.** *A deterministic verifier $\mathsf{V}^* = \{\mathsf{V}^*_\lambda\}_\lambda$ is always explainable if for any prover P^*, $\lambda \in \mathbb{N}$, $x \in \{0,1\}^\lambda$, and $T = \langle \mathsf{P}^*, \mathsf{V}^*_\lambda\rangle(x)$, T is explainable and is not $\perp$.*

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 (w.l.o.g). We note that in the two-message setting, we can assume w.l.o.g that explainable verifiers are *always* explainable. This is because aborts can be easily simulated and malicious verifier messages (that are not abort) occur with negligible probability.

2.1.4 Witness Indistinguishability

We consider two-message witness-indistinguishable (WI) arguments with delayed input, where the prover obtains the input statement and witness after the first round of the protocol. In some of our protocols, we will also require a weak argument of knowledge property that says that there exists a quasipolynomial-time witness extractor.

Definition 2.7 (Two-message argument with delayed input)**.** *A two-message protocol* $\langle \mathsf{P}, \mathsf{V} \rangle$ *is a delayed input WI argument if* V *consists of two PPT algorithms* $(\mathsf{V}_1, \mathsf{V}_2)$ *that satisfy:*

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

$$\Pr\left[\mathsf{V}_2(x, \mathsf{wi}_2; \tau) = 1 \;\middle|\; \begin{array}{l} (\mathsf{wi}_1, \tau) \leftarrow \mathsf{V}_1(1^\lambda) \\ \mathsf{wi}_2 \leftarrow \mathsf{P}(x, w, \mathsf{wi}_1) \end{array}\right] = 1 \ .$$

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* $\lambda \in \mathbb{N}$*,*

$$\Pr\left[\begin{array}{l} \mathsf{V}_2(x, \mathsf{wi}_2; \tau) = 1 \\ x \in \{0,1\}^\lambda \setminus \mathcal{L} \end{array} \;\middle|\; \begin{array}{l} (\mathsf{wi}_1, \tau) \leftarrow \mathsf{V}_1(1^\lambda) \\ (x, \mathsf{wi}_2) \leftarrow \mathsf{P}^*(\mathsf{wi}_1) \end{array}\right] \leq \mu(\lambda) \ .$$

3. **Witness indistinguishability:** *For any polynomial-size verifier* $\mathsf{V}^* = \{\mathsf{V}^*_\lambda\}_\lambda$*, there exists a negligible* μ *such that for any security parameter* $\lambda \in \mathbb{N}$*,*

$$\Pr\left[\begin{array}{l} \mathsf{V}^*_\lambda(\mathsf{wi}_2) = b \\ x \in \mathcal{L} \cap \{0,1\}^\lambda \\ w_0, w_1 \in \mathcal{R}_\mathcal{L}(x) \end{array} \;\middle|\; \begin{array}{l} (\mathsf{wi}_1, x, w_0, w_1) \leftarrow \mathsf{V}^*_\lambda \\ b \leftarrow \{0,1\} \\ \mathsf{wi}_2 \leftarrow \mathsf{P}(x, w_b, \mathsf{wi}_1) \end{array}\right] \leq \frac{1}{2} + \mu(\lambda) \ .$$

The argument has a quasipolynomial witness extractor *if there exists a quasipolynomial time extractor* E *such that for any polynomial-size prover* $\mathsf{P}^* = \{\mathsf{P}^*_\lambda\}_\lambda$*, there exists a negligible* μ *such that for any security parameter* $\lambda \in \mathbb{N}$*,*

$$\Pr\left[\begin{array}{l} \mathsf{V}_2(x, \mathsf{wi}_2; \tau) = 1 \\ w \notin \mathcal{R}_\mathcal{L}(x) \end{array} \;\middle|\; \begin{array}{l} (\mathsf{wi}_1, \tau) \leftarrow \mathsf{V}_1(1^\lambda) \\ (x, \mathsf{wi}_2) \leftarrow \mathsf{P}^*(\mathsf{wi}_1) \\ w \leftarrow \mathsf{E}(x, \mathsf{wi}_1, \mathsf{wi}_2) \end{array}\right] \leq \mu(\lambda) \ .$$

Instantiations. Two-message WI arguments with delayed input that are also 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].

A two-message WI argument with delayed input and a quasipolynomial witness extractor can be constructed from any (plain) two-message WI argument and subexponentially-secure two-message commitments [Pas03], which in turn can be constructed from subexponentially-secure one-way functions [Nao91, HILL99].

2.2 Commitments

Non-interactive bit commitments. We define bit commitments.

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

$$\mathsf{Com} : \{0,1\} \times \{0,1\}^{\lambda} \to \{0,1\}^{\ell(\lambda)}$$

is a bit commitment if it satisfies:

1. **Binding:** *For any $r, r' \in \{0,1\}^{\lambda}, b, b' \in \{0,1\}$, 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 $\lambda \in \mathbb{N}$,*

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

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

The commitment is **dense** *if for any string $s \in \{0,1\}^{\ell(\lambda)}$ there exist (b,r) such that $s = \mathsf{Com}(b;r)$.*

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

Extractable commitments. We define 3-message extractable commitment schemes.

Definition 2.9. *A 3-message extractable commitment scheme $(\mathsf{EC.S}, \mathsf{EC.R}, \mathsf{EC.V})$ satisfies*

1. **Indistinguishability:** *For any polynomial-size receiver $\mathsf{R}^* = \{\mathsf{R}^*_\lambda\}_{\lambda\in\mathbb{N}}$, there exists a negligible μ such that for any security parameter $\lambda \in \mathbb{N}$, and any two equal-length s_0, s_1,*

$$\Pr\left[\mathsf{R}^*_\lambda(\mathsf{c}_1,\mathsf{c}_2,\mathsf{c}_3) = b \;\middle|\; \begin{array}{r} b \leftarrow \{0,1\} \\ (\mathsf{c}_1,\tau) \leftarrow \mathsf{EC.S}(s_b) \\ \mathsf{c}_2 \leftarrow \mathsf{R}^*(\mathsf{c}_1) \\ \mathsf{c}_3 \leftarrow \mathsf{EC.S}(\mathsf{c}_1,\mathsf{c}_2;\tau) \end{array}\right] \le \frac{1}{2} + \mu(\lambda) \enspace .$$

2. **Extraction:** *There exists a PPT extractor E, such that for any deterministic sender $\mathsf{S}^* = \{\mathsf{S}^*_\lambda\}_\lambda$ and any security parameter $\lambda \in \mathbb{N}$,*

$$\textit{if } \Pr\left[\mathsf{EC.V}(\mathsf{c}_1,\mathsf{c}_2,\mathsf{c}_3) = 1 \;\middle|\; \begin{array}{r} \mathsf{c}_1 \leftarrow \mathsf{S}^* \\ \mathsf{c}_2 \leftarrow \mathsf{EC.R}(\mathsf{c}_1, 1^\lambda) \\ \mathsf{c}_3 \leftarrow \mathsf{S}^*(\mathsf{c}_1,\mathsf{c}_2) \end{array}\right] \ge \varepsilon \enspace ,$$

$$\textit{then } \Pr\left[\begin{array}{l} \mathsf{EC.V}(\mathsf{c}_1,\mathsf{c}_2,\mathsf{c}_3) = 1 \\ \exists s : (\mathsf{c}_1,\mathsf{c}_2,\mathsf{c}_3) \in \mathsf{EC.S}(s) \\ s' \leftarrow \mathsf{E}^{\mathsf{S}^*}(1^\lambda, 1^{1/\varepsilon}) \\ s' \ne s \end{array} \;\middle|\; \begin{array}{r} \mathsf{c}_1 \leftarrow \mathsf{S}^* \\ \mathsf{c}_2 \leftarrow \mathsf{EC.R}(\mathsf{c}_1, 1^\lambda) \\ \mathsf{c}_3 \leftarrow \mathsf{S}^*(\mathsf{c}_1,\mathsf{c}_2) \end{array}\right] \le 2^{-\lambda} \enspace ,$$

where $(\mathsf{c}_1,\mathsf{c}_2,\mathsf{c}_3) \in \mathsf{EC.S}(s)$ denotes the fact that the transcript $(\mathsf{c}_1,\mathsf{c}_2,\mathsf{c}_3)$ is consistent with an honest commitment to a string s.

The scheme is **2-message with a quasipolynomial extractor** *if c_1 is always empty, and E runs in quasipolynomial time.*

Remark 2.7 (2-Message extractable commitments). In the two message case, the extraction guarantee can be simplified to require that whenever $(\mathsf{c}_2,\mathsf{c}_3) \in \mathsf{EC.S}(s)$, the extractor outputs s. This, in particular, implies the above definition, which will be sufficient for us.

Instantiation. 3-Message extractable commitments can be constructed from non-interactive commitments [PRS02]. 2-Message extractable commitments with a quasipolynomial extractor can be constructed from subexponential one-way functions [Nao91, HILL99, CGGM00].

2.3 Fully-Homomorphic Encryption

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

Definition 2.10. *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,
$$\mathsf{FHE.Dec}_{\mathsf{sk}}(\mathsf{FHE.Eval}(C, \mathsf{FHE.Enc}_{\mathsf{sk}}(m))) = C(m) \enspace .$$

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,
$$\mathsf{FHE.Enc}_{\mathsf{sk}}(m_0) \approx_{\mathsf{D}_\lambda, \mu} \mathsf{FHE.Enc}_{\mathsf{sk}}(m_1) \enspace ,$$
where $\mathsf{FHE.Enc}_{\mathsf{sk}}(m_b)$ *is the distribution of encryptions of* m_b *with random secret key* $\mathsf{sk} \leftarrow \{0,1\}^\lambda$.

3. **Compactness:** *There exists a fixed polynomial* poly, *such that for any* $\lambda \in \mathbb{N}, \mathsf{sk} \in \{0,1\}^\lambda$, *message* $m \in \{0,1\}^*$, *and circuit* C,
$$|\mathsf{FHE.Eval}(C, \mathsf{FHE.Enc}_{\mathsf{sk}}(m))| \leq \mathrm{poly}(|C(m)|, \lambda) \enspace .$$

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].[5]

2.4 Compute and Compare Obfuscation

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

Definition 2.11 (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 2.12 (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* $f(x) = u$,
$$\Pr\left[\widetilde{\mathbf{CC}}(x) = 1 \;\middle|\; \widetilde{\mathbf{CC}} \leftarrow \mathcal{O}(\mathbf{CC}[f,u])\right] = 1 \enspace .$$

2. **Simulation:** *there exists a PPT simulator* Sim *such that*
   - *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* $f : \{0,1\}^n \to \{0,1\}^\lambda$,
$$\mathcal{O}(\mathbf{CC}[f,u]) \approx_{\mathsf{D}_\lambda, \mu} \mathsf{Sim}(1^\lambda, 1^\ell) \enspace ,$$
     where $u \leftarrow \{0,1\}^\lambda$ *is chosen uniformly at random.*

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

- *Simulated circuits are rejecting:*

$$\Pr\left[\exists x : \widetilde{\mathbf{CC}}(x) = 1 \,\middle|\, \widetilde{\mathbf{CC}} \leftarrow \mathsf{Sim}(1^\lambda, 1^\ell)\right] \leq 2^{-\lambda} \ .$$

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 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 classic 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 2.13. *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* $\lambda \in \mathbb{N}$,

$$\mathsf{pk}, \mathsf{RSR.Enc}_{\mathsf{pk}}(0) \approx_{\mathsf{D}_\lambda, \mu} \mathsf{pk}, \mathsf{RSR.Enc}_{\mathsf{pk}}(1) \ ,$$

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 (possibly probabilistic) distinguisher* D *and* ε,

   - *if*

$$\left|\mathbb{E}\big(\mathsf{D}(\mathsf{RSR.Enc}_{\mathsf{pk}}(0))\big) - \mathbb{E}\big(\mathsf{D}(\mathsf{RSR.Enc}_{\mathsf{pk}}(1))\big)\right| \geq \varepsilon \ ,$$

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

$$\Pr\left[\mathsf{RSR.}\widetilde{\mathsf{Dec}}^{\mathsf{D}}(\mathsf{ct}, \mathsf{pk}, 1^{1/\varepsilon}) = b\right] = 1 - 2^{-\lambda} \ ,$$

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

Remark 2.8 (Self reducibility against unbounded distinguishers). In the above definitions of random self-reduction, the distinguisher D is allowed to be unbounded. This is not essential in our constructions, but does make our proof cleaner, and is satisfied by all considered instantiations.

Relaxed RSR. We may consider a relaxed version of RSR encryption, where ciphertexts $\mathsf{ct} \in \mathsf{RSR.Enc_{pk}}(b)$ 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 (possibly probabilistic) distinguisher D and ε,
   - if
   $$\left|\mathbb{E}\mathsf{D}(\mathsf{RSR.}\widetilde{\mathsf{Enc}}_{\mathsf{pk}}(0)) - \mathbb{E}\mathsf{D}(\mathsf{RSR.}\widetilde{\mathsf{Enc}}_{\mathsf{pk}}(1))\right| \geq \varepsilon \enspace ,$$
   - then for any $b \in \{0,1\}$ and $\mathsf{ct} \in \mathsf{RSR.Enc_{pk}}(b)$,
   $$\Pr\left[\mathsf{RSR.}\widetilde{\mathsf{Dec}}^{\mathsf{D}}(\mathsf{ct}, \mathsf{pk}, 1^{1/\varepsilon}) = b\right] = 1 - 2^{-\lambda} \enspace ,$$
   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. Assuming quasipolynomial hardness of the underlying problems they are also quasipolynomially secure.

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 give relaxed RSR schemes using the standard noise flooding technique [Gen09a] (see Appendix A).

2.6 Witness Encryption

We recall the definition of witness encryption (WE).

Definition 2.14. *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* $m \in \{0,1\}^*$*,*
   $$\mathsf{WE.Dec}_w(\mathsf{WE.Enc}_x(m))) = m \enspace .$$
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*,*
   $$\mathsf{WE.Enc}_x(m_0) \approx_{\mathsf{D}_\lambda,\mu} \mathsf{WE.Enc}_x(m_1) \enspace ,$$
   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 for any language in **NP**. State of the art schemes for **NP** (which haven't been broken) include constructions based on indistinguishability obfuscation [GGH+16] or the GGH15 multi-linear maps [CVW18]. Witness encryption is also known for any language that has a *hash proof system* [CS02].

2.7 Conditional Disclosure of Secrets

Conditional disclosure of secrets for an **NP** language $\mathcal{L}$ [AIR01, BP12, AJ17] can be viewed as a two-message analog of witness encryption. That is, the sender holds an instance x and message m and the receiver holds x and a corresponding witness w. If the witness is valid, then the receiver obtains m, whereas if $x \notin \mathcal{L}$, m remains hidden. We further require that the protocol hides the witness w from the sender.

Definition 2.15. *A conditional disclosure of secrets scheme* $(\mathsf{CDS.R}, \mathsf{CDS.S}, \mathsf{CDS.D})$ *for a language* $\mathcal{L} \in$ ***NP*** *satisfies:*

1. **Correctness:** *for any* $(x, w) \in \mathcal{R}_{\mathcal{L}}$*, and message* $m \in \{0,1\}^*$*,*

$$\Pr\left[\mathsf{CDS.D}_{\mathsf{k}}(\mathsf{ct}_{\mathsf{S}}) = m \;\middle|\; \begin{array}{l} (\mathsf{ct}_{\mathsf{R}}, \mathsf{k}) \leftarrow \mathsf{CDS.R}(x, w) \\ \mathsf{ct}_{\mathsf{S}} \leftarrow \mathsf{CDS.S}(x, m, \mathsf{ct}_{\mathsf{R}}) \end{array}\right] = 1 \enspace .$$

2. **Message 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* $x \in \{0,1\}^\lambda \setminus \mathcal{L}$*,* $\mathsf{ct}_{\mathsf{R}}^*$*, and two equal-length messages* m_0, m_1*,*

$$\mathsf{CDS.S}(x, m_0, \mathsf{ct}_{\mathsf{R}}^*) \approx_{\mathsf{D}_\lambda, \mu} \mathsf{CDS.S}(x, m_1, \mathsf{ct}_{\mathsf{R}}^*) \enspace .$$

3. **Receiver simulation:** *There exists a simulator* $\mathsf{CDS.Sim}$*, such that 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* $x \in \mathcal{L}$*, and* $w \in \mathcal{R}_{\mathcal{L}}(x)$*,*

$$\mathsf{ct}_{\mathsf{R}} \approx_{\mathsf{D}_\lambda, \mu} \mathsf{CDS.Sim}(x) \enspace ,$$

where $\mathsf{ct}_{\mathsf{R}} \leftarrow \mathsf{CDS.R}(x, w)$.

Instantiations. CDS schemes can be instantiated assuming any two-message oblivious transfer protocol where the receiver message is computationally hidden from any semi-honest sender, and with (unbounded) simulation security against malicious receivers. Such oblivious transfer schemes are known based on DDH [NP01], Quadratic (or N^{th}) Residuosity [HK12], and LWE [BD18].

3 Weak Zero-Knowledge against Explainable Verifiers

In this section, we construct WZK protocols against explainable verifiers. We start with the three-message protocol and then move to the 2-message protocol, which will be a special case of the 3-message protocol.

3.1 The Three-Message Protocol

In this section, we construct a three-message WZK argument against explainable verifiers.

Ingredients and notation:

- A 3-message extractable commitment $(\mathsf{EC.S}, \mathsf{EC.R}, \mathsf{EC.V})$. We denote its messages by $(\mathsf{c}_1, \mathsf{c}_2, \mathsf{c}_3)$.
- a 2-message WI argument $(\mathsf{WI.P}, \mathsf{WI.V})$ 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.

Remark 3.1 (FHE Compactness). In the protocol the decryption circuit $\mathsf{FHE.Dec_{sk}}$ is meant to decrypt encryptions of messages of size $|u| = \lambda$. Such encryptions could be the result of homomorphic evaluation, and according to the compactness property of the FHE will have fixed polynomial size in the security parameter.

3.1.1 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. Assume toward contradiction that there exists a polynomial-size (w.l.o.g deterministic) prover $\mathsf{P}^* = \{\mathsf{P}^*_\lambda\}$ that (for infinitely many $\lambda \in \mathbb{N}$) breaks soundness with noticeable probability $\varepsilon(\lambda) = \lambda^{-O(1)}$. Fix $\lambda \in \mathbb{N}$ and $x \in \{0,1\}^\lambda \setminus \mathcal{L}$ such that P^*_λ convinces the verifier of accepting with probability ε.

Augmenting the interaction with extraction. We parse any verifier message $(\mathsf{c}_2, \mathsf{wi}_1, \mathsf{cmt}, \mathsf{ct_V}, \widetilde{\mathbf{CC}}, \mathsf{ct}'_\mathsf{V}, \mathsf{pk}')$, as two parts (c_2, z). For any z, we consider a sender S^*_z whose first message is the same c_1 output by P^*_λ, and given a message c_2 from the receiver $\mathsf{EC.R}$, computes the third message c_3 by running P^*_λ on (c_2, z), emulating a message form V. For any transcript $T = (\mathsf{c}_1, (\mathsf{c}_2, z), (\mathsf{c}_3, \mathsf{ct_P}, \mathsf{wi}_2))$ of an execution between P^*_λ and V, we consider the result s_T of running the witness extractor $s_T \leftarrow \mathsf{E}^{\mathsf{S}^*_z}(1^\lambda, 1^{4/\varepsilon})$. (Note that s_T is a random variable that may depend on T and the extractor's coins.)

Let F be the event, over a transcript T and coins of the extractor E, that:

1. V successfully verifies the commitment transcript $(\mathsf{c}_1, \mathsf{c}_2, \mathsf{c}_3)$,
2. V accepts the WI argument $(\mathsf{wi}_1, \mathsf{wi}_2)$,
3. the the extracted string s_T does **not** attest that $\mathsf{ct_P}$ is a zero encryption; that is, s_T is **not** randomness r' such that $\mathsf{ct_P} = \mathsf{RSR.Enc}(0; r')$.

We next consider several hybrid experiments and show that the probability that F occurs is preserved through these experiments, upto a negligible difference. To reach a contradiction, we show that F occurs in the first hybrid with probability at least ε and with probability at most $\varepsilon/2$ in the last hybrid. This is a contradiction since the gap $\varepsilon/2$ is noticeable. In what follows, in each hybrid i, we will denote by p_i the probability that F occurs in that hybrid.

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

Claim 3.1. $p_0 \geq \varepsilon$.

Proof. By definition, whenever V accepts, it decrypts $\mathsf{ct_P}$ to 1; thus, by the correctness of RSR, the ciphertext $\mathsf{ct_P}$ cannot be opened to an encryption of 0. The claim now follows from the fact that P^*_λ convinces V with probability ε. □

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

Claim 3.2. $|p_0 - p_1| \leq \lambda^{-\omega(1)}$.

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. P computes $(\mathsf{c}_1, \tau) \leftarrow \mathsf{EC.S}(0^\lambda)$, the first message and state of an extractable commitment to zero. It sends c_1.

2. V computes

   - $\mathsf{c}_2 \leftarrow \mathsf{EC.R}(\mathsf{c}_1)$, the second message of the extractable commitment.
   - $(\mathsf{wi}_1, \tau_\mathsf{V}) \leftarrow \mathsf{WI.V}_1(1^\lambda)$, the first WI message and a corresponding state.
   - $\mathsf{cmt} \leftarrow \mathsf{Com}(0; r)$, a commitment to zero, using randomness $r \leftarrow \{0,1\}^\lambda$,
   - $\mathsf{ct}_\mathsf{V} \leftarrow \mathsf{FHE.Enc}_\mathsf{sk}(r)$, 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 $u \leftarrow \{0,1\}^\lambda$.
   - $\mathsf{ct}'_\mathsf{V} \leftarrow \mathsf{RSR.Enc}_\mathsf{pk}(u)$, an RSR encryption of the target u, where $(\mathsf{sk}', \mathsf{pk}') \leftarrow \mathsf{RSR.Gen}(1^\lambda)$ are randomly chosen keys.

   It sends $(\mathsf{c}_2, \mathsf{wi}_1, \mathsf{cmt}, \mathsf{ct}_\mathsf{V}, \widetilde{\mathbf{CC}}, \mathsf{ct}'_\mathsf{V}, \mathsf{pk}')$.

3. P computes

   - $\mathsf{c}_3 \leftarrow \mathsf{EC.S}(\mathsf{c}_1, \mathsf{c}_2; \tau)$, the third message of the extractable commitment.
   - $\mathsf{ct}_\mathsf{P} \leftarrow \mathsf{RSR.Enc}_{\mathsf{pk}'}(1)$, an RSR encryption of 1.
   - $\mathsf{wi}_2 \leftarrow \mathsf{WI.P}(\Psi, w, \mathsf{wi}_1)$, the second WI message for the statement

$$
\begin{aligned}
&\Psi(x, \mathsf{cmt}, \widetilde{\mathbf{CC}}, \mathsf{ct}_\mathsf{P}, \mathsf{pk}', \mathsf{c}_1, \mathsf{c}_2, \mathsf{c}_3) := \\
&\qquad \exists w \ : \ (x, w) \in \mathcal{R}_\mathcal{L} && \bigvee \\
&\qquad \exists r \ : \ \mathsf{cmt} = \mathsf{Com}(0; r) && \bigvee \\
&\qquad \exists \widehat{\mathsf{ct}} \ : \ \widetilde{\mathbf{CC}}(\widehat{\mathsf{ct}}) = 1 && \bigvee \\
&\qquad \exists r' \ : \ \mathsf{ct}_\mathsf{P} = \mathsf{RSR.Enc}_{\mathsf{pk}'}(0; r') \ \bigwedge \ (\mathsf{c}_1, \mathsf{c}_2, \mathsf{c}_3) \in \mathsf{EC.S}(r') && .
\end{aligned}
$$

   It sends $(\mathsf{c}_3, \mathsf{ct}_\mathsf{P}, \mathsf{wi}_2)$.

4. V verifies

   - the commitment $\mathsf{EC.V}(\mathsf{c}_1, \mathsf{c}_2, \mathsf{c}_3) = 1$,
   - that the WI argument is accepted: $\mathsf{WI.V}_2(\Psi, \mathsf{wi}_1, \mathsf{wi}_2; \tau_\mathsf{V}) = 1$,
   - that the prover's ciphertext decrypts to one: $\mathsf{RSR.Dec}_{\mathsf{sk}'}(\mathsf{ct}_\mathsf{P}) = 1$.

Figure 1: A 3-message WZK argument for **NP** against explainable verifiers.

Proof. The claim follows by the semantic security of the scheme RSR. Indeed, given a noticeable difference between p_0 and p_1, we can distinguish an RSR encryption of a random u from one of 0^λ, by emulating an interaction between P^*_λ and V, running the extractor, and testing whether F occurs. □

$\boldsymbol{\mathcal{H}}_2$**:** 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 $\mathsf{FHE.Dec}_{\mathsf{sk}}$.

Claim 3.3. $|p_1 - p_2| \leq \lambda^{-\omega(1)}$.

Proof. The claim follows by the CC simulation guarantee. Indeed, given a noticeable difference between p_1 and p_2, we can distinguish, given sk, an obfuscation of $\mathbf{CC}[\mathsf{FHE.Dec}_{\mathsf{sk}}, u]$ for a random u, from a simulated obfuscation contradicting the CC guarantee. Using the fact that now $\mathsf{ct}'_{\mathsf{V}}$ is an encryption of 0^λ, we can again emulate an interaction between P^*_λ and V. □

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

Claim 3.4. $|p_2 - p_3| \leq \lambda^{-\omega(1)}$.

Proof. The claim follows by the semantic security of the FHE scheme FHE. Indeed, given a noticeable difference between p_2 and p_3, we can distinguish, an encryption of a random r from an encryption of 0^λ. Since now the FHE key sk is not needed to compute the obfuscation $\widetilde{\mathbf{CC}}$, we can again emulate an interaction between P^*_λ and V. □

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

Claim 3.5. $|p_3 - p_4| \leq \lambda^{-\omega(1)}$.

Proof. The claim follows by hiding of the commitment. Given a noticeable difference between p_3 and p_4, we can distinguish, a commitment to zero from a commitment to one. In this hybrid, ct_{V} no longer depends on the commitment randomness, so we can again emulate an interaction between P^*_λ and V. □

.

It is left to show that in $\boldsymbol{\mathcal{H}}_4$, F occurs with probability at most $\varepsilon/2$.

Claim 3.6. $p_4 \leq \varepsilon/2$.

Proof. Assume toward contradiction that $p_4 > \varepsilon/2$. First, observe that in hybrid $\boldsymbol{\mathcal{H}}_4$:

- $x \notin \mathcal{L}$.
- $\mathsf{cmt} \in \mathsf{Com}(1)$, 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 $2^{-\lambda}$, there does not exist $\widehat{\mathsf{ct}}$ such that $\widetilde{\mathbf{CC}}(\widehat{\mathsf{ct}}) = 1$.

In particular, we can fix the second part z of the verifier's message such that with probability at least $\varepsilon/2 - 2^{-\lambda}$, over c_2:

1. The above three conditions hold (for $\mathsf{cmt}, \widetilde{\mathbf{CC}}$ fixed by z).
2. The WI verifier WI.V accepts the argument (for the statement Ψ).
3. The commitment verifier EC.V accepts the transcript $(\mathsf{c}_1, \mathsf{c}_2, \mathsf{c}_3)$.
4. The extractor $\mathsf{E}^{\mathsf{S}^*_z}(1^\lambda, 1^{4/\varepsilon})$ outputs s_T that does not attest that ct_{P} is a zero encryption.

Invoking the soundness of the WI argument, it follows that with probability at least $\varepsilon/2 - 2^{-\lambda}\lambda^{-\omega(1)} \gg 2^{-\lambda}$, over c_2:

1. The commitment verifier $\mathsf{EC.V}$ accepts the transcript $(\mathsf{c}_1, \mathsf{c}_2, \mathsf{c}_3)$.
2. $\exists r'$ such that $(\mathsf{c}_1, \mathsf{c}_2, \mathsf{c}_3) \in \mathsf{EC.S}(r')$, and r' attests that ct_P is a zero encryption.
3. The extractor $\mathsf{E}^{\mathsf{S}^*_z}(1^\lambda, 1^{4/\varepsilon})$ outputs s_T that does not attest that ct_P is a zero encryption. In particular, $s_T \neq r'$.

This contradicts the extraction guarantee of the commitment. □

This concludes the proof of soundness. □

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 malicious verifier V^* outputs its view consisting of its random coins and prover messages. (Otherwise, we 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.)

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

1. Sample coins r^* for V^*_λ. Henceforth, $\mathsf{V}^*_{r^*}$ denotes the corresponding (deterministic) verifier.
2. Sample a random $r' \leftarrow \{0,1\}^\lambda$ and a first commitment message and state $(\mathsf{c}_1, \tau) \leftarrow \mathsf{EC.S}(r')$ corresponding to a commitment to r'. Store the randomness r_c used to generate the commitment. Feed c_1 to $\mathsf{V}^*_{r^*}$.
3. If the verifier aborts, output $(r^*, \mathsf{c}_1, \bot)$.
4. Obtain $(\mathsf{c}_2, \mathsf{cmt}, \mathsf{ct}_\mathsf{V}, \widetilde{\mathbf{CC}}, \mathsf{ct}'_\mathsf{V}, \mathsf{pk}', \mathsf{wi}_1)$ from $\mathsf{V}^*_{r^*}$
5. Compute the third commitment message $\mathsf{c}_3 \leftarrow \mathsf{EC.S}(\mathsf{c}_1, \mathsf{c}_2; \tau)$.
6. Construct the (*homomorphic simulation*) circuit $\mathsf{HS}(r)$ that given an input r:
   - Construct a distinguisher $\mathsf{D}'(\mathsf{ct}_\mathsf{P})$ for the RSR encryption that given a ciphertext ct_P:
     - Samples at random a second WI message
       $$\mathsf{wi}_2 \leftarrow \mathsf{WI.P}(\Psi, r, \mathsf{wi}_1)$$
       for the statement $\Psi(x, \mathsf{cmt}, \widetilde{\mathbf{CC}}, \mathsf{ct}_\mathsf{P}, \mathsf{pk}', \mathsf{c}_1, \mathsf{c}_2, \mathsf{c}_3)$, using as the witness the randomness r attesting that $\mathsf{cmt} = \mathsf{Com}(0; r)$.
     - Runs $\mathsf{D}_\lambda(r^*, \mathsf{c}_1, (\mathsf{c}_3, \mathsf{ct}_\mathsf{P}, \mathsf{wi}_2))$.
   - Applies the decryptor
     $$\tilde{u} \leftarrow \mathsf{RSR}.\widetilde{\mathsf{Dec}}^{\mathsf{D}'}(\mathsf{ct}'_\mathsf{V}, \mathsf{pk}', 1^{1/\varepsilon}) \enspace ,$$
     and output $\tilde{u}$.

   All randomness required by the above is hardwired into HS.
7. Compute $\widehat{\mathsf{ct}} = \mathsf{FHE.Eval}(\mathsf{HS}, \mathsf{ct}_\mathsf{V})$.
8. If $\widetilde{\mathbf{CC}}(\widehat{\mathsf{ct}}) = 1$:

- Sample $\mathsf{ct_P} \leftarrow \mathsf{RSR.Enc_{pk'}}(1)$.
- Sample a second WI message

$$\mathsf{wi}_2 \leftarrow \mathsf{WI.P}(\Psi, \widehat{\mathsf{ct}}, \mathsf{wi}_1)$$

for the statement $\Psi(x, \mathsf{cmt}, \widetilde{\mathbf{CC}}, \mathsf{ct_P}, \mathsf{pk}', \mathsf{c}_1, \mathsf{c}_2, \mathsf{c}_3)$, using as the witness $\widehat{\mathsf{ct}}$ attesting that $\widetilde{\mathbf{CC}}(\widehat{\mathsf{ct}}) = 1$.

9. Otherwise:

- Compute $\mathsf{ct_P} = \mathsf{RSR.Enc_{pk'}}(0; r')$, where r' is the randomness underlying the extractable commitment.
- Sample a second WI message

$$\mathsf{wi}_2 \leftarrow \mathsf{WI.P}(\Psi, (r', r_{\mathsf{c}}), \mathsf{wi}_1)$$

for the statement $\Psi(x, \mathsf{cmt}, \widetilde{\mathbf{CC}}, \mathsf{ct_P}, \mathsf{pk}', \mathsf{c}_1, \mathsf{c}_2, \mathsf{c}_3)$, using as the witness the randomness r' attesting that $\mathsf{ct_P} \leftarrow \mathsf{RSR.Enc_{pk'}}(0; r')$ and the commitment randomness r_{c}.

10. Output $(r^*, \mathsf{c}_1, (\mathsf{c}_3, \mathsf{ct_P}, \mathsf{wi}_2))$.

Simulation validity. The simulator clearly runs in polynomial time (in its input length). We focus on proving validity. 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 $\varepsilon(\lambda) = \lambda^{-O(1)}$.

We consider a sequence of hybrid simulators that transition from the simulator to the prover, and prove that the views generated by each two consecutive hybrids are indistinguishable.

S_0**:** This is the real simulator S.

S_1**:** This simulator is the same as S, only that in Step 9, if it needs to give an argument (due to unsuccessful extraction of a proper $\widehat{\mathsf{ct}}$), it uses the witness w, instead of (r', r_{c}).

S_2**:** This simulator is the same as S_1, only that instead of an extractable commitment $(\mathsf{c}_1, \mathsf{c}_2, \mathsf{c}_3)$ to r', it provides a commitment to 0^λ.

S_3**:** This simulator is inefficient. It acts the same as S_2, only that:

- It checks whether, the verifier's message is explainable; namely, in the support of the honest verifier messages (or $\perp$), and aborts if its not.
- Then it finds the commitment randomness r, and uses it to provide proofs both in Step 8 and in Step 9.

P_1**:** This simulator is also inefficient. It acts the same as the honest prover P, only that:

- It checks whether, the verifier's message is explainable; namely, in the support of the honest verifier messages (or $\perp$), and aborts if its not.
- Then it finds the commitment randomness r, and uses it to provide the WI argument.

P_0**:** This simulator emulates the real prover.

To deduce the validity of the simulator, we prove:

Claim 3.7. *There exists a negligible $\nu(\lambda)$ such that for all $\lambda \in \mathbb{N}$, $x \in \mathcal{L} \cap \{0,1\}^\lambda$, $w \in \mathcal{R}_{\mathcal{L}}(x)$, $i \in [3]$,*

$$\begin{aligned}\mathsf{S}_{i-1}(x, \mathsf{V}^*_\lambda, \mathsf{D}_\lambda, 1^{1/\varepsilon}) &\approx_{\mathsf{D}_\lambda,\nu} \mathsf{S}_i(x, \mathsf{V}^*_\lambda, \mathsf{D}_\lambda, 1^{1/\varepsilon}) \ , \\ \mathsf{S}_3(x, \mathsf{V}^*_\lambda, \mathsf{D}_\lambda, 1^{1/\varepsilon}) &\approx_{\mathsf{D}_\lambda,\varepsilon+2^{-\lambda}} \mathsf{OUT}_{\mathsf{V}^*_\lambda}\langle \mathsf{P}_1(w), \mathsf{V}^*_\lambda\rangle(x) \ , \\ \mathsf{OUT}_{\mathsf{V}^*_\lambda}\langle \mathsf{P}_1(w), \mathsf{V}^*_\lambda\rangle(x) &\approx_{\mathsf{D}_\lambda,\nu} \mathsf{OUT}_{\mathsf{V}^*_\lambda}\langle \mathsf{P}(w), \mathsf{V}^*_\lambda\rangle(x) \ .\end{aligned}$$

Proof. We prove the indistinguishability of each two consecutive hybrid simulators.

$\underline{\mathsf{S} \approx \mathsf{S}_1}$: Recall that the only difference between these two simulators is that S_1 uses the witness w in Step 9, instead of (r', r_c). Assume toward contradiction that D distinguishes the views generated by the two simulators with noticeable probability $\delta(\lambda)$. We use D to construct a verifier $\mathsf{WI.V}^*$ that breaks witness indistinguishability of the underlying WI argument.

$\mathsf{WI.V}^*$ runs S to sample a transcript, but discards the WI argument wi_2 that S produces. Instead, it outputs as the first WI message the message wi_1 produced by V^* during the simulation, as well as the statement Ψ and $w_0 = (r', r_\mathsf{c})$ and $w_1 = w$ as the two witnesses. It then receives a WI second message wi'_2 generated using one of these witnesses w_i. It then runs D on the corresponding view $(r^*, \mathsf{c}_1, (\mathsf{c}_3, \mathsf{ct}'_\mathsf{P}, \mathsf{wi}'_2)$, which is identical to that generated by S, except that wi_2 generated by S is replaced by the wi'_2 received from the challenger. $\mathsf{WI.V}^*$ outputs whatever D outputs.

It is left to observe that for each $i \in \{0,1\}$, the generated view is distributed identically to that generated by S_i. It follows that $\mathsf{WI.V}^*$ has advantage δ in the WI challenge.

$\underline{\mathsf{S}_1 \approx \mathsf{S}_2}$: Recall that the only difference between these two simulators is that S_2 commits to 0^λ instead of to r' like S_1. Assume toward contradiction that D distinguishes the views generated by the two simulators with noticeable probability $\delta(\lambda)$, then we construct a receiver R^* that breaks the indistinguishability of the extractable commitment with the same advantage δ.

R^* emulates S_1, when sampling r', R^* submits to the challenger $s_1 = r'$ and $s_2 = 0^\lambda$. Then, instead of sampling c_1 as part of the simulation, R^* obtains it from an outside challenger. It proceeds with the emulation of S_1, feeding c_1 to V^*, and obtaining c_2 as part of V^*'s message. R^* sends c_2 to the challenger and obtains c_3. It then performs the rest of the emulation using c_3 (note that S_1 no longer requires the randomness underlying the commitment in Step 9).

It is left to observe that for each $i \in \{1,2\}$, when the challenger commits to s_i, the generated view is distributed identically to that generated by S_i. It follows that R^* has advantage δ in the WI challenge.

$\underline{\mathsf{S}_2 \approx \mathsf{S}_3}$: Recall that the difference between these two simulators is that S_3 aborts if the verifier's message is not explainable, and if it is uses the commitment randomness r as the witness in Steps 8 and 9, instead of $\widehat{\mathsf{ct}}$ and w, respectively. Assume toward contradiction that D distinguishes the views generated by the two simulators with noticeable probability $\delta(\lambda)$. We use D to construct a verifier $\mathsf{WI.V}^*$ that breaks witness indistinguishability of the underlying WI argument. While the simulator S_3 is inefficient, $\mathsf{WI.V}^*$ will be efficient (albeit non-uniform).

First note that the first messages c_1 of the two simulators are distributed identically. Furthermore, since V^* is explainable, there exists a negligible $\mu(\lambda)$ such that except with probability μ over the choice of first message and randomness r^* for V^*_λ, the verifier's message is explainable. Thus by averaging there exists a fixed first message c_1 and randomness r^* such that conditioned on (c_1, r^*), D_λ distinguishes S from S_1 with advantage $\delta - \mu$ and the verifier's message is explainable. We fix such (c_1, r^*) non-uniformly, which also fixes the randomness r, underlying the verifier's V^* commitment (we can assume w.l.o.g that the verifiers message is not $\perp$, because then both simulators abort).

The verifier $\mathsf{WI.V}^*$ has (c_1, r^*, r) hardwired, as well as the state τ corresponding to c_1. It emulates S_2 starting from the point it obtains the message from V^* (Step 4), obtains a corresponding transcript, but discards the WI argument wi_2 that S_2 produces. Instead:

- If the transcript reaches Step 8 (corresponding to successful extraction of $\widehat{\mathsf{ct}}$), $\mathsf{WI.V}^*$ sends the challenger the first WI message wi_1 generated by V^* along with the statement Ψ and witnesses $w_2 = \widehat{\mathsf{ct}}$ and $w_3 = r$.

- If the transcript reaches Step 9 (corresponding to unsuccessful extraction of $\widehat{\mathsf{ct}}$), $\mathsf{WI.V}^*$ sends to the challenger wi_1 and Ψ with witnesses $w_2 = w$ and $w_3 = r$.

It then receives a WI second message wi'_2 generated using one of these witnesses w_i. It then runs D on the corresponding view $(r^*, \mathsf{c}_1, (\mathsf{c}_3, \mathsf{ct}'_{\mathsf{P}}, \mathsf{wi}'_2)$, which is identical to that generate by S_2, except that wi_2 generated by S_2 is replaced by the wi'_2 received from the challenger. $\mathsf{WI.V}^*$ outputs whatever D outputs.

It is left to observe that for each $i \in \{2, 3\}$, the generated view is distributed identically to that generated by S_i. It follows that $\mathsf{WI.V}^*$ has advantage $\delta - \mu$ in the WI challenge.

$\underline{\mathsf{P}_1 \approx \mathsf{P}}$: We jump to prove this indistinguishability as it is very similar to the previous one (the tired reader is advised to skip it).

Recall that the difference between these two provers is that P_1 aborts if the verifier's message is not explainable, and if it is, uses the commitment randomness r as the witness when proving Ψ instead of w like P. Assume toward contradiction that D distinguishes the views generated by the two provers with noticeable probability $\delta(\lambda)$. We use D to construct a verifier $\mathsf{WI.V}^*$ that breaks witness indistinguishability of the underlying WI argument. While the prover P_1 is inefficient, $\mathsf{WI.V}^*$ will be efficient (albeit non-uniform).

First note that the first messages c_1 of the two provers are distributed identically. Furthermore, since V^* is explainable, there exists a negligible $\mu(\lambda)$ such that except with probability μ over the choice of first message and randomness r^* for V^*_λ, the verifier's message is explainable. Thus by averaging there exists a fixed first message c_1 and randomness r^* such that conditioned on (c_1, r^*), D_λ distinguishes P from P_1 with advantage $\delta - \mu$ and the verifier's message is explainable. We fix such (c_1, r^*) non-uniformly, which also fixes the randomness r, underlying the verifier's V^* commitment (we can assume w.l.o.g that the verifiers message is not $\perp$, because then both simulators abort).

The verifier $\mathsf{WI.V}^*$ has (c_1, r^*, r) hardwired, as well as the state τ corresponding to c_1. It emulates P starting from the point it obtains the message from V^* (the third protocol message), obtains a corresponding transcript, but discards the WI argument wi_2 that P_1 produces. Instead, it sends the challenger the first WI message wi_1 generated by V^* along with the statement Ψ and witnesses $w_0 = w$ and $w_1 = r$. It then receives a WI second message wi'_2 generated using one of these witnesses w_i. It then runs D on the corresponding view $(r^*, \mathsf{c}_1, (\mathsf{c}_3, \mathsf{ct}'_{\mathsf{P}}, \mathsf{wi}'_2)$, which is identical to that generate by P, except that wi_2 generated by S_2 is replaced by the wi'_2 received from the challenger. $\mathsf{WI.V}^*$ outputs whatever D outputs.

It is left to observe that for each $i \in \{0, 1\}$, the generated view is distributed identically to that generated by P_i. It follows that $\mathsf{WI.V}^*$ has advantage $\delta - \mu$ in the WI challenge.

$\underline{\mathsf{S}_3 \approx \mathsf{P}_1}$: Note that up to receiving the verifier's V^* message, and adding the generation of c_3, the views generated by the two simulators are distributed identically. Thus, to prove $(\mathsf{D}_\lambda, \varepsilon + 2^{-\lambda})$-indistinguishability of these two simulators, it suffices to prove $(\mathsf{D}_\lambda, \varepsilon + 2^{-\lambda})$-indistinguishability conditioned on any fixing of the first prover message, verifier randomness r^* and message, and c_3. Note that if the verifier's message is not explainable, both simulators abort, and thus we concentrate on the case that the verifier's message is explainable. From hereon consider such a fixing and let r be the corresponding randomness.

Let Δ be the advantage of the probabilistic 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 $\mathsf{ct}'_\mathsf{V} \in \mathsf{RSR.Enc}_{\mathsf{pk}'}(u)$, due to explainability. Then by the random self reducibility of RSR, with overwhelming probability $1 - 2^{-\lambda}$, over the coins of $\mathsf{RSR}.\widetilde{\mathsf{Dec}}$, it holds that

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

in which case, $\mathsf{HS}(r) = u$. Assume that this is indeed the case.

By the correctness of FHE, the ciphertext $\widehat{\mathsf{ct}} = \mathsf{FHE.Eval}(\mathsf{HS}, \widehat{\mathsf{ct}})$ obtained by the simulator S_3 satisfies:

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

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

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

This corresponds to successful extraction of $\widehat{\mathsf{ct}}$, and will result in the simulator S_3 behaving identically to the prover P_1 (in Step 8) — sampling ct_P as a one encryption and wi_2 using the witness r.

Thus, in case 1, we have $(\mathsf{D}_\lambda, 2^{-\lambda})$-indistinguishability.

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 simulator behaves exactly like P_1. Henceforth, we assume that the simulator does not obtain such a witness, in which case it reaches Step 9. Here the difference between the simulator S_3 and prover P_1 is that the first samples $\mathsf{ct}_\mathsf{P} \leftarrow \mathsf{RSR.Enc}_{\mathsf{pk}'}(0)$, whereas the second samples $\mathsf{ct}_\mathsf{P} \leftarrow \mathsf{RSR.Enc}_{\mathsf{pk}'}(1)$. The advantage of D_λ in distinguishing the two is exactly the advantage $\Delta \leq \varepsilon$ of D'. This completes the proof of this case, and of Claim 3.7. □

Remark 3.2 (On using relaxed random self-reducible encryption). We can rely on relaxed RSR encryption by slightly tweaking the protocol. Specifically, in the protocol, the encryption ct_P will be under the alternative RSR encryption algorithm $\mathsf{RSR}.\widetilde{\mathsf{Enc}}$. The simulation and analysis remain the same.

3.2 The Two-Message Protocol

We also construct, under stronger assumptions, a two-message WZK argument against explainable verifiers and quasipolynomial provers. Specifically we strengthen the ingredients as follows.

Ingredients and notation:

- A **2-message** extractable commitment $(\mathsf{EC.S}, \mathsf{EC.R}, \mathsf{EC.V})$ with a quasipolynomial extractor. We denote its messages by $(\mathsf{c}_2, \mathsf{c}_3)$; this is for consistency with the three-message protocol (c_1 is always empty).
- a 2-message WI argument $(\mathsf{WI.P}, \mathsf{WI.V})$ with delayed input, **sound against quasipolynomial provers.**
- A non-interactive perfectly-binding commitment scheme Com, **hiding against qausipolynomial distinguishers.**
- A fully-homomorphic encryption scheme $(\mathsf{FHE.Enc}, \mathsf{FHE.Dec}, \mathsf{FHE.Eval})$, **secure against qausipolynomial distinguishers.**
- A compute-and-compare obfuscator $\mathcal{O}$, **secure against qausipolynomial distinguishers.**

- A random self-reducible public-key encryption $(\mathsf{RSR.Gen}, \mathsf{RSR.Enc}, \mathsf{RSR.Dec}, \mathsf{RSR.}\widetilde{\mathsf{Dec}})$, **secure against qausipolynomial distinguishers.** (In fact, relaxed RSR suffices. To simplify the description of the protocol, we rely on standard RSR, and later remark why relaxed RSR suffices.)

The protocol is identical to the one in Figure 1, only that since the first commitment message c_1 is always empty, it consists of only two messages.

Proposition 3.3. *Protocol Figure 1, instantiated with the above ingredients, is a 2-message WZK protocol against explainable verifiers and soundness against quasipolynomial provers.*

Proof sketch. The proof is essentially identical to that of the previous protocol, only taking into account the above enhancements.

In the proof of soundness, the prover P^* and the extractor E are now quasipolynomial instead of polynomial. Thus, each of the reductions in the proof is now quasipolynomial instead of polynomial, and breaks the quasipolynomial (rather than polynomial) security of the underlying primitives.

The same WZK proof applies. In fact, it can be simplified. This is because obtaining the commitment randomness r can now be done efficiently (non-uniformly), which means that we can invoke the indistinguishability of the extractable commitment in the presence of any one of the simulators. Thus instead of first switching to simulating the argument in the simulation Step 9 with w, we can directly switch to r. □

4 Witness Hiding against Explainable Verifiers with Public Verification

In this section, relying on witness encryption, we give a two-message protocol that is also publicly-verifiable. The new protocol, however, is only WH. 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 $(\mathsf{WI.P}, \mathsf{WI.V})$. 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 $(\mathsf{WE.Enc}, \mathsf{WE.Dec})$ for languages in **NP**.

We describe the protocol in Figure 2.

Remark 4.1 (FHE Compactness). Remark 3.1 regarding the fixed size of the circuit $\mathsf{FHE.Dec}_{\mathsf{sk}}$ holds for the above protocol as well.

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.*

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 $r \leftarrow \{0,1\}^\lambda$,
   - $\mathsf{ct}_\mathsf{V} \leftarrow \mathsf{FHE.Enc}_\mathsf{sk}(r)$, 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 $u \leftarrow \{0,1\}^\lambda$.
   - $\mathsf{ct}'_\mathsf{V} \leftarrow \mathsf{WE.Enc}_x(u)$, a witness encryption of the target u.

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

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 $\mathcal{L}$.

The soundness analysis is a simplification of that of Protocol 1. 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, transitioning from the real system to a system where no prover cannot convince the verifier of accepting. We show that the probability that the prover convinces the verifier to accept is preserved throughout the hybrids. Since the argument is publicly verifiable, it suffices to show that the prover's view is indistinguishable between these hybrids.

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

$\mathcal{H}_1$**:** In this hybrid, ct'_V 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_V 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 $2^{-\lambda}$, there does not exist $\widehat{\mathsf{ct}}$ such that $\widetilde{\mathbf{CC}}(\widehat{\mathsf{ct}}) = 1$.

Overall we deduce that, with overwhelming probability $1 - 2^{-\lambda}$, the statement $\Psi(x, \mathsf{cmt}, \widetilde{\mathbf{CC}})$ is false. By the soundness of the WI argument, the prover cannot cheat in this hybrid except with negligible probability. □

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

Proof. We describe the witness-finding reduction R.

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

- Obtain $(\mathsf{wi}_1, \mathsf{cmt}, \mathsf{ct}_\mathsf{V}, \widetilde{\mathbf{CC}}, \mathsf{ct}'_\mathsf{V})$ from V^*_λ.
- Construct the (*homomorphic simulation*) circuit $\mathsf{HS}(r)$ that given an input r:
  - Sample 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 $\mathsf{cmt} = \mathsf{Com}(0; r)$.
  - Feed wi_2 to V^*_λ and obtain a candidate witness $\tilde{w}$.
  - Apply the witness decryptor
  $$\tilde{u} \leftarrow \mathsf{WE.Dec}_{\tilde{w}}(\mathsf{ct}'_\mathsf{V}) \ ,$$
  and output $\tilde{u}$.

  All randomness required by the above is hardwired to HS.
- Compute $\widehat{\mathsf{ct}} = \mathsf{FHE.Eval}(\mathsf{HS}, \mathsf{ct}_\mathsf{V})$.
- 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 $\perp$.

Reduction validity. The reduction clearly runs in polynomial time. We now prove its validity for always-explainable verifiers (which is w.l.o.g, Remark 2.6). Let $\mathsf{V}^* = \{\mathsf{V}^*_\lambda\}_\lambda$ be any always-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. First, we argue that when HS gives V^*_λ the second WI message wi_2, V^*_λ outputs a witness w with probability $\delta - \lambda^{-\omega(1)}$. Otherwise, we can construct a verifier $\mathsf{WI.V}^*$ that breaks witness indistinguishability. The verifier $\mathsf{WI.V}^*_\lambda$ has the witness r non-uniformly hardwired. It obtains V^*_λ's message, and gives the challenger the WI first message wi_1 generated by V^*_λ, along with the statement Ψ, and two witnesses $w_0 = w$ and $w_1 = r$, when it receives wi_2, it uses it to complete the emulation of V^*_λ. It then check if V^*_λ outputs a witness for x, if it does $\mathsf{WI.V}^*_\lambda$ outputs 0, and otherwise outputs 1. By construction the advantage of $\mathsf{WI.V}^*$ is exactly the difference between the probabilities of outputting a witness.

Next, note that whenever the verifier V^* outputs a witness (under the encryption), the witness decryption operation performed by HS, 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 $\delta - \lambda^{-\omega(1)}$. Otherwise, we can again construct $\mathsf{WI.V}^*$ that will break witness indistinguishability similarly to the previous reduction, but now using the (non-uniformly hardwired) . 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 From Explainable Verifiers to Malicious Ones

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

This includes the following:

- A 3-message transformation that preserves WZK (or ZK), based on polynomial hardness assumptions.
- A 2-message transformation that preserves WZK, based on super-polynomial hardness assumptions.
- A 2-message transformation that preserves WH, based on polynomial witness encryption.

5.1 The Three-Message Transformation

We provide a transformation that compiles any 3-message WZK protocol against explainable verifiers into one against malicious verifiers.

Ingredients and notation:

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

- A non-interactive dense commitment scheme Com.

- A 3-message argument system $\langle \mathsf{P}, \mathsf{V} \rangle$ for an **NP** language $\mathcal{L}$ that is WZK against explainable verifiers. 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.
   - $\mathsf{cmt} \leftarrow \mathsf{Com}(w)$ a commitment to the witness.
   - wi_1, the first message of a WI argument.

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

2. $\bar{\mathsf{V}}$ computes

   - arg_2, the second message in the original protocol.
   - wi_2, the second WI message for the statement $\Phi(x, \mathsf{cmt}, \mathsf{arg}_1, \mathsf{arg}_2)$:

   $$\exists r \ : \ \mathsf{arg}_2 = \mathsf{V}(x, \mathsf{arg}_1; r) \ \bigvee \ \exists r, s \ : \ \mathsf{cmt} = \mathsf{Com}(s; r), s \notin \mathcal{R}_\mathcal{L}(x) \ .$$

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

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, and sends it.

4. $\bar{\mathsf{V}}$ verifies the argument $(\mathsf{arg}_1, \mathsf{arg}_2, \mathsf{arg}_3)$.

Figure 3: A WZK 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 $\mathsf{P}^* = \{\mathsf{P}^*_\lambda\}_\lambda$, and show that for any $x \in \{0,1\}^\lambda \setminus \mathcal{L}$, if $\bar{\mathsf{P}}^*_\lambda$ convinces $\bar{\mathsf{V}}$ to accept with probability ε, the new prover convinces P^* convinces V to accept with probability $\varepsilon - \lambda^{-\omega(1)}$. For this, consider the first message $(\mathsf{arg}_1, \mathsf{cmt}, \mathsf{wi}_1)$ sent by $\bar{\mathsf{P}}^*_\lambda$. Since the commitment Com is dense, there exists an underlying string s and randomness r, such that $\mathsf{cmt} = \mathsf{Com}(s; r)$. The constructed P^*_λ will have (s, r) non-uniformly hardwired into its code, and will operate as follows.

P^*_λ:

- It sends arg_1 to V, and obtains arg_2.
- It computes the argument message wi_2 for $\Phi(x, \mathsf{cmt}, \mathsf{arg}_1, \mathsf{arg}_2)$ using the witness (r, s).
- It then feeds $(\mathsf{arg}_2, \mathsf{wi}_2)$ to $\bar{\mathsf{P}}^*_\lambda$, and obtains back arg_3. It sends arg_3 to V.

Prover analysis. P^* is clearly of polynomial size. We now analyze its cheating probability. First, 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 $\Phi(x, \mathsf{cmt}\, \mathrm{arg}_1, \mathrm{arg}_2)$, 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 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. *Assume that the original protocol $\langle \mathsf{P}, \mathsf{V}\rangle$ is WZK against explainable verifiers, then Protocol 3 is WZK.*

Proof. Fix any polynomial-size (malicious) verifier $\bar{\mathsf{V}}^* = \left\{\bar{\mathsf{V}}^*_\lambda\right\}_\lambda$ against Protocol 3 and a polynomial-size distinguisher $\mathsf{D} = \{\mathsf{D}_\lambda\}_\lambda$. To prove that the protocol is WZK, we first prove that $\bar{\mathsf{V}}^*$ can be converted into an explainable verifier against the original protocol, which is given as auxiliary input a commitment to the witness.

Claim 5.1. *There exists a PPT simulator* E *such that:*

- $\mathsf{V}^* = \left\{\mathsf{V}^*_{x,\mathsf{cmt}} := \mathsf{E}^{\bar{\mathsf{V}}^*_\lambda}(x, \mathsf{cmt})\right\}_{x,\mathsf{cmt}}$ *is an explainable verifier against the original protocol* $\langle \mathsf{P}, \mathsf{V}\rangle$ *for all $x \in \mathcal{L} \cap \{0,1\}^\lambda$, $w \in \mathcal{R}_\mathcal{L}(x)$, and* $\mathsf{cmt} \in \mathsf{Com}(w)$.
- *For any $\lambda \in \mathbb{N}$, $x \in \mathcal{L} \cap \{0,1\}^\lambda$, and $w \in \mathcal{R}_\mathcal{L}(x)$,*
$$\mathsf{OUT}_{\bar{\mathsf{V}}^*_\lambda}\langle \mathsf{P}(w), \bar{\mathsf{V}}^*_\lambda\rangle(x) \equiv \mathsf{OUT}_{\mathsf{V}^*_{x,\mathsf{cmt}}}\langle \mathsf{P}(w), \mathsf{V}^*_{x,\mathsf{cmt}}\rangle(x) \ ,$$
where $\mathsf{cmt} \leftarrow \mathsf{Com}(w)$.

Before proving the claim let us show that it implies that the protocol is WZK. For this purpose, we describe the corresponding WZK simulator.

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

- Sample a commitment of zero $\mathsf{cmt} \leftarrow \mathsf{Com}(0^{|w|})$.
- Construct the verifier $\mathsf{V}^*_{x,\mathsf{cmt}} := \mathsf{E}^{\bar{\mathsf{V}}^*_\lambda}(x, \mathsf{cmt})$.
- Output $\mathsf{S}(x, \mathsf{V}^*_{x,\mathsf{cmt}}, 1^{1/\varepsilon})$, where S is the simulator of the original protocol $\langle \mathsf{P}, \mathsf{V}\rangle$.

To prove the validity of $\bar{\mathsf{S}}$, we consider an alternative $\bar{\mathsf{S}}_w$ that acts similarly, except that it samples a commitment of the witness $\mathsf{cmt} \leftarrow \mathsf{Com}(w)$, rather than a commitment of zero. Then by the hiding of the commitment

$$\bar{\mathsf{S}}(x, \bar{\mathsf{V}}^*_\lambda, \mathsf{D}_\lambda, 1^{1/\varepsilon}) \approx_{\mathsf{D}_\lambda, \lambda^{-\omega(1)}} \bar{\mathsf{S}}_w(x, \bar{\mathsf{V}}^*_\lambda, \mathsf{D}_\lambda, 1^{1/\varepsilon}) \ .$$

Furthermore, by construction

$$\bar{\mathsf{S}}_w(x, \bar{\mathsf{V}}^*_\lambda, \mathsf{D}_\lambda, 1^{1/\varepsilon}) \equiv \mathsf{S}(x, \bar{\mathsf{V}}_{x,\mathsf{cmt}}, \mathsf{D}_\lambda, 1^{1/\varepsilon}) \ ,$$

where $\mathsf{cmt} \leftarrow \mathsf{Com}(w)$. By the first part of Claim 5.1, the corresponding verifier $\bar{\mathsf{V}}_{x,\mathsf{cmt}}$ is explainable, and thus by the WZK guarantee of the original protocol $\langle \mathsf{P}, \mathsf{V} \rangle$,

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

By the second part of Claim 5.1, we deduce

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

as required.

Proof of Claim 5.1. We describe the explainable simulator E.

$\mathsf{V}^*_{x,\mathsf{cmt}} := \mathsf{E}^{\bar{\mathsf{V}}^*_\lambda}(x, \mathsf{cmt})$**:**

- Given prover message arg_1 from P, sample a first WI message wi_1, feed $(\mathsf{arg}_1, \mathsf{cmt}, \mathsf{wi}_1)$ to $\bar{\mathsf{V}}^*_\lambda$, and obtain its message $(\mathrm{arg}_2, \mathsf{wi}_2)$.
- Verify the WI argument $(\mathsf{wi}_1, \mathsf{wi}_2)$ for the statement $\Phi(x, \mathsf{cmt}, \mathsf{arg}_{1},_{2}\ )$.
- If verification does not pass, emulate an abort:
  - Send P message $\perp$.
  - Feed $\bar{\mathsf{V}}^*_\lambda$ with a message $\perp$ (emulating an abort of $\bar{\mathsf{P}}$), and output whatever $\bar{\mathsf{V}}^*_\lambda$ does.
- Otherwise, send arg_2 to P, obtain arg_3, feed it to $\bar{\mathsf{V}}^*_\lambda$, and output whatever $\bar{\mathsf{V}}^*_\lambda$ does.

First, by the construction of E, for any $\lambda \in \mathbb{N}$, $x \in \mathcal{L} \cap \{0,1\}^\lambda$, and $w \in \mathcal{R}_\mathcal{L}(x)$,

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

We now prove that $\mathsf{V}^* = \left\{ \mathsf{V}^*_{x,\mathsf{cmt}} := \mathsf{E}^{\bar{\mathsf{V}}^*_\lambda}(x, \mathsf{cmt}) \right\}_{x,\mathsf{cmt}}$ is explainable for all $x \in \mathcal{L} \cap \{0,1\}^\lambda$, $w \in \mathcal{R}_\mathcal{L}(x)$, and $\mathsf{cmt} \in \mathsf{Com}(w)$.

Assume toward contradiction that when interacting with P, V^*_λ outputs a message that is not explainable with noticeable probability $\delta(\lambda)$. We use V^* to construct a prover wiP^* that breaks the soundness of the WI argument. Given a first WI message wi_1, wiP^*_λ emulates V^*_λ, but replaces the message wi_1 that V^*_λ generates with the one received from the challenger. It then obtains from the emulated V^*_λ, the statement $\Phi(x, \mathsf{cmt}, \mathsf{arg}_1, \mathsf{arg}_2)$ and second WI message wi_2, it returns (Φ, wi_2) to the challenger. By construction and the assumption that V^*_λ's message is not explainable with probability δ, wiP^* breaks soundness with probability exactly δ.

This completes the proof of the claim. □

□

Remark 5.1 (Zero knowledge). We note that the above transformation holds just the same for ZK — the constructed explainable verifier V^* only depends on the malicious verifier $\bar{\mathsf{V}}^*$, and not on the distinguisher.

5.2 The Two-Message Transformation

We provide a transformation that compiles any 2-message WZK protocol against explainable verifiers with soundness against quasipolynomial provers (like the one from Section 3.2) into one against malicious verifiers.

Ingredients and notation:

- A 2-message WI argument for **NP** with delayed input with a quasipolynomial witness extractor $\langle \mathsf{WI.P}, (\mathsf{WI.V}_1, \mathsf{WI.V}_2)\rangle$. We denote its messages by $(\mathsf{wi}_1, \mathsf{wi}_2)$.

- A conditional disclosure of secrets scheme $(\mathsf{CDS.R}, \mathsf{CDS.S}, \mathsf{CDS.D})$ for **NP**, with receiver simulation security against quasi-polynomial time senders.

- A 2-message argument system $\langle \mathsf{P}, \mathsf{V}\rangle$ for an **NP** language $\mathcal{L}$ that is WZK against explainable verifiers and sound against quasipolynomial provers. We denote its messages by $(\mathsf{arg}_1, \mathsf{arg}_2)$.

We describe the protocol in Figure 4.

Analysis. We now analyze the transformation.

Proposition 5.3. *Protocol 4 is sound.*

Proof. To prove soundness, we transform any cheating prover $\bar{\mathsf{P}}^*$ against Protocol 4 into a quasi-polynomial cheating prover P^* against the original protocol $\langle \mathsf{P}, \mathsf{V}\rangle$. We describe P^*.

P^*_λ**:**

- Obtain the first message arg_1 from the verifier V.

- Simulate the CDS receiver first message $\mathsf{ct}_\mathsf{R} = \mathsf{CDS.Sim}(\Psi)$, relative to the statement $\Psi(\mathsf{arg}_1)$ attesting that arg_1 is honest.

- Sample a first WI message wi_1, feed $(\mathsf{wi}_1, \mathsf{arg}_1, \mathsf{ct}_\mathsf{R})$ to $\bar{\mathsf{P}}^*_\lambda$, and obtain $(\mathsf{ct}_\mathsf{S}, \mathsf{wi}_2)$.

- Apply the quasipolynomial witness extractor $\mathsf{arg}_2 \leftarrow \mathsf{E}(\Phi, \mathsf{wi}_1, \mathsf{wi}_2)$ for the statement $\Phi(x, \mathsf{arg}_1, \mathsf{ct}_\mathsf{S}, \mathsf{ct}_\mathsf{R})$.

- Send the extracted arg_2 to V.

Prover analysis. Fix any polynomial-size prover $\bar{\mathsf{P}}^* = \{\bar{\mathsf{P}}^*_\lambda\}_\lambda$. First note that the corresponding new prover P^* runs in quasipolynomial time. We show that for any $x \notin \mathcal{L}$, if $\bar{\mathsf{P}}^*_\lambda$ convinces $\bar{\mathsf{V}}$ to accept with noticeable probability $\varepsilon(\lambda)$, the new prover P^*_λ convinces V to accept with probability $\varepsilon - \lambda^{-\omega(1)}$.

First, we consider a hybrid experiment where the prover strategy $\widetilde{\mathsf{P}}^*$ is identical to that of P^*, except that instead of sampling a simulated CDS receiver message $\mathsf{ct}_\mathsf{R} \leftarrow \mathsf{CDS.Sim}(\Psi)$ for the statement $\Psi(\mathsf{arg}_1)$, $\widetilde{\mathsf{P}}^*$ obtains externally a CDS message $\mathsf{ct}_\mathsf{R} \leftarrow \mathsf{CDS.R}(\Psi, r)$ corresponding to the randomness used by V to sample arg_1. ($\widetilde{\mathsf{P}}^*$ then uses ct_R like P^*).

Claim 5.2. $\widetilde{\mathsf{P}}^*$ *convinces* V *with the same probability as* P^* *does, up to a negligible difference.*

Proof. Otherwise, we can use $\widetilde{\mathsf{P}}^*$ to construct a quasipolynomial distinguisher D that breaks the receiver simulation property. D_λ emulates an interaction between $\widetilde{\mathsf{P}}^*_\lambda$ and V. It then submits $\Psi(\mathsf{arg}_1)$ and r to a challenger, where arg_1 is the message generated by V^* using randomness r. It receives back ct_R, and completes the emulation. It is left to note that if $\mathsf{ct}_\mathsf{R} \leftarrow \mathsf{CDS.Sim}(\Psi)$, then the view of V is distributed as in an interaction with P^*_λ, whereas if $\mathsf{ct}_\mathsf{R} \leftarrow \mathsf{CDS.S}(\Psi, r)$, the view is distributed as in an interaction with $\widetilde{\mathsf{P}}^*_\lambda$. □

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

   - $(\mathsf{wi}_1, \tau_{\mathsf{V}}) \leftarrow \mathsf{WI.V}_1(1^\lambda)$, the first WI message.
   - arg_1, the verifier message in the original protocol $\langle \mathsf{P}, \mathsf{V} \rangle(x)$. It stores the randomness r used by the verifier to generate the message.
   - $(\mathsf{ct}_{\mathsf{R}}, \mathsf{k}) \leftarrow \mathsf{CDS.R}(\Psi, r)$, a CDS receiver message for the statement $\Psi(\mathsf{arg}_1)$ attesting that arg_1 was computed according to the honest verifier V, using r as the witness.

   It sends $(\mathsf{wi}_1, \mathsf{arg}_1, \mathsf{ct}_{\mathsf{R}})$.

2. $\bar{\mathsf{P}}$ computes

   - arg_2, the prover message in the original protocol $\langle \mathsf{P}, \mathsf{V} \rangle$.
   - $\mathsf{ct}_{\mathsf{S}} \leftarrow \mathsf{CDS.S}(\Psi, \mathsf{arg}_2, \mathsf{ct}_{\mathsf{R}})$, a CDS sender message encrypting arg_2.
   - $\mathsf{wi}_2 \leftarrow \mathsf{WI.P}(\Phi, \mathsf{arg}_2, \mathsf{wi}_1)$, the second WI message for the statement

   $$\Phi(x, \Psi, \mathsf{ct}_{\mathsf{S}}, \mathsf{ct}_{\mathsf{R}}) := \left(\exists \mathsf{arg} \ : \ \mathsf{ct}_{\mathsf{S}} \in \mathsf{CDS.S}(\Psi, \mathsf{arg}, \mathsf{ct}_{\mathsf{R}}) \right) \bigvee \left(x \in \mathcal{L} \right) .$$

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

3. $\bar{\mathsf{V}}$ then

   - Runs $\mathsf{WI.V}_2(\Phi, \mathsf{wi}_1, \mathsf{wi}_2; \tau_{\mathsf{V}})$ to verify the WI argument for the statement Φ.
   - Decrypts $\widetilde{\mathsf{arg}}_2 \leftarrow \mathsf{CDS.D}_{\mathsf{k}}(\mathsf{ct}_{\mathsf{S}})$.
   - Verifies the original argument $(\mathsf{arg}_1, \widetilde{\mathsf{arg}}_2)$.

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

From hereon we focus on proving that $\widetilde{\mathsf{P}}^*$ convinces V of accepting with probability $\varepsilon - \lambda^{-\omega(1)}$. Let arg_1 be the message received from V, let ct_{S} be sender encryption emulated by $\widetilde{\mathsf{P}}^*_\lambda$, let arg_2 be the prover message that $\widetilde{\mathsf{P}}^*_\lambda$ extracts from $\bar{\mathsf{P}}^*_\lambda$, and let $\widetilde{\mathsf{arg}}_2 = \mathsf{CDS.D}_{\mathsf{k}}(\mathsf{ct}_{\mathsf{S}})$ be the decrypted message with respect to the secret key k produced when generating the receiver message ct_{R}. Also let $(\mathsf{wi}_1, \mathsf{wi}_2)$ be the WI argument for the statement $\Phi(x, \Psi, \mathsf{ct}_{\mathsf{R}}, \mathsf{ct}_{\mathsf{S}})$ generated by $\bar{\mathsf{P}}^*_\lambda$ during its emulation by $\widetilde{\mathsf{P}}^*_\lambda$.

Claim 5.3. *With probability at least $\varepsilon - \lambda^{-\omega(1)}$,*

1. V *accepts* $(\mathsf{arg}_1, \widetilde{\mathsf{arg}}_2)$.
2. $\widetilde{\mathsf{arg}}_2 = \mathsf{arg}_2$.

This implies that $\widetilde{\mathsf{P}}^*$, who sends arg_2, convinces V with probability at least $\varepsilon - \lambda^{-\omega(1)}$, as required and would complete the proof.

Proof of Claim 5.3. By construction, an interaction between $\widetilde{\mathsf{P}}^*$ and V perfectly emulates an interaction between $\bar{\mathsf{P}}^*$ and $\bar{\mathsf{V}}$. In such an interaction the verifier $\bar{\mathsf{V}}$ both accepts the WI argument $(\mathsf{wi}_1, \mathsf{wi}_2)$ for

$\Phi(x, \Psi, \mathsf{ct}_\mathsf{R}, \mathsf{ct}_\mathsf{S})$ and accepts the underlying $(\mathsf{arg}_1, \widetilde{\mathsf{arg}}_2)$. Since $x \notin \mathcal{L}$, it follows by the extraction guarantee, that except with negligible probability $\lambda^{-\omega(1)}$, ct_S is a valid CDS encryption of arg_2. Furthermore, by CDS correctness, it holds that $\widetilde{\mathsf{arg}}_2 = \mathsf{CDS.D}_\mathsf{k}(\mathsf{ct}_\mathsf{S}) = \mathsf{arg}_2$. □

This completes the proof of souneness.

□

Proposition 5.4. *Protocol 4 is weak zero-knowledge against malicious verifiers.*

Proof. We describe a simulator $\bar{\mathsf{S}}$. Throughout, we assume w.l.o.g that the simulated verifier $\bar{\mathsf{V}}^* = \{\bar{\mathsf{V}}^*_\lambda\}_\lambda$ is deterministic and always outputs the prover message it receives (Remark 2.4).

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

- Obtain $(\mathsf{wi}_1, \mathsf{arg}_1, \mathsf{ct}_\mathsf{R})$ from $\bar{\mathsf{V}}^*_\lambda$.
- Construct a new verifier V^*_λ that sends arg_1 as its first message, and given arg_2 from P, outputs it.
- Construct a new distinguisher D_λ that given arg_2 from P:
  - Samples $\mathsf{ct}_\mathsf{S} \leftarrow \mathsf{CDS.S}(\Psi, \mathsf{arg}_2, \mathsf{ct}_\mathsf{R})$, an encryption of arg_2 under $\Psi(\mathsf{arg}_1)$.
  - Samples $\mathsf{wi}_2 \leftarrow \mathsf{WI.P}(\Phi, (\mathsf{arg}_2, r_\mathsf{cds}), \mathsf{wi}_1)$, a second WI message for the statement $\Phi(x, \Psi, \mathsf{ct}_\mathsf{R}, \mathsf{ct}_\mathsf{S})$, using as the witness the message arg_2 and randomness r_cds used for generating ct_S.
  - Runs $\bar{\mathsf{D}}_\lambda(\mathsf{ct}_\mathsf{S}, \mathsf{wi}_2)$.
- Obtain $\widetilde{\mathsf{arg}}_2 \leftarrow \mathsf{S}(x, \mathsf{V}^*_\lambda, \mathsf{D}_\lambda, 1^{1/\varepsilon})$, where S is the simulator for the protocol $\langle \mathsf{P}, \mathsf{V} \rangle$.
- Compute a CDS encryption $\widetilde{\mathsf{ct}}_\mathsf{S} \leftarrow \mathsf{CDS.S}(\Psi, \widetilde{\mathsf{arg}}_2, \mathsf{ct}_\mathsf{R})$.
- Compute a second WI message $\widetilde{\mathsf{wi}}_2 \leftarrow \mathsf{WI.P}(\Phi, (\widetilde{\mathsf{arg}}_2, \tilde{r}_\mathsf{cds}), \mathsf{wi}_1)$, using as the witness the message $\widetilde{\mathsf{arg}}_2$ and randomness $\tilde{r}_\mathsf{cds}$ used for generating $\widetilde{\mathsf{ct}}_\mathsf{S}$.
- Output $(\widetilde{\mathsf{ct}}_\mathsf{S}, \widetilde{\mathsf{wi}}_2)$.

Simulator analysis. The simulator $\bar{\mathsf{S}}$ clearly runs in polynomial time. We now prove its validity.

Assume toward contradiction that there exist polynomial-size distinguisher $\bar{\mathsf{D}} = \{\bar{\mathsf{D}}_\lambda\}_\lambda$ and verifier $\bar{\mathsf{V}}^* = \{\bar{\mathsf{V}}^*_\lambda\}_\lambda$ that for infinitely many $x \in \mathcal{L}$ and $w \in \mathcal{R}_\mathcal{L}(x)$ distinguishes $\bar{\mathsf{S}}(x, \bar{\mathsf{V}}^*_\lambda, \bar{\mathsf{D}}_\lambda, 1^{1/\varepsilon})$ from $\mathsf{OUT}_{\bar{\mathsf{V}}^*_\lambda}\langle \bar{\mathsf{P}}^*(w), \bar{\mathsf{V}}^*_\lambda \rangle(x)$ with advantage $\varepsilon(\lambda) + \delta(\lambda)$ for noticeable ε, δ. We consider two cases.

Case 1: There exists a set H of infinitely many x as above such that the verifier's message arg_1 is in the support of the honest verifier's messages. We show that this contradicts WZK against explainable verifiers.

By the WZK guarantee of $\langle \mathsf{P}, \mathsf{V} \rangle$ against explainable verifiers, there exists a negligible $\mu(\lambda)$ such that for any $x \in H \cap \{0,1\}^\lambda$,

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

Furthermore, by the definition of $\bar{\mathsf{S}}, \mathsf{D}$, for any such x,

$$\mathsf{D}_\lambda(\mathsf{S}(x, \mathsf{V}^*_\lambda, \mathsf{D}_\lambda, 1^{1/\varepsilon})) \equiv \bar{\mathsf{D}}_\lambda(\bar{\mathsf{S}}(x, \bar{\mathsf{V}}^*_\lambda, \bar{\mathsf{D}}_\lambda, 1^{1/\varepsilon})) \enspace .$$

Finally, by the definition of $\mathsf{D}, \mathsf{V}^*, \mathsf{P}$, for any such x,

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

It follows that for any $x \in H \cap \{0,1\}^{\lambda}$,

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

This disproves Case 1.

Case 2: There exists a set M of infinitely many x as above such that the verifier's message arg_1 is malicious (not in the support of the honest verifier's messages). We show how to use $\bar{\mathsf{V}}^*$, , to break the CDS message hiding. First we consider an alternative simulator $'$ that has the witness $w \in \mathcal{R}_{\mathcal{L}}(x)$ hardwired. It computes $\widetilde{\mathsf{wi}_2}$ in the simulation using the witness w instead of using as the witness $\widetilde{\mathsf{arg}_2}, \tilde{r}_{\mathsf{cds}}$. Similarly, we consider an alternative prover $\bar{\mathsf{P}}'$, which computes its own message wi_2 using w instead of arg_2 and r_{cds}. By witness indistinguishability:

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

Thus $\bar{\mathsf{D}}$ distinguishes the view $(\mathsf{CDS.S}(\Psi, \widetilde{\mathsf{arg}_2}, \mathsf{ct_R})), \widetilde{\mathsf{wi}_2}(w)$ generated by $\bar{\mathsf{S}}'$ from $(\mathsf{CDS.S}(\Psi, \mathsf{arg}_2, \mathsf{ct_R})), \mathsf{wi}_2(w)$ generated by P' with advantage $\varepsilon + \delta - \lambda^{-\omega(1)}$. However, since the verfier's message is malicious $\Psi(\mathsf{arg}_1)$ is false. Thus in this case we obtain a distinguisher against the CDS message hiding.

This completes the proof of the proposition. □

5.3 The Two-Message WH Transformation

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 3, where the verifier, rather than using a WI system to prove that it behaves honestly, proves it using a witness encryption of its coins under x. This proof is simulatable if $x \notin \mathcal{L}$, and otherwise is sound.

We describe the protocol in Figure 5.

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 5 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.

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.

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 $\bar{\mathsf{V}}^* = \{\bar{\mathsf{V}}^*_\lambda\}_\lambda$ be a polynomial-size verifier.

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

- 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 $w \in \mathcal{R}_\mathcal{L}(x)$, and if so outputs it.

- Otherwise, constructs from $\bar{\mathsf{V}}^*_\lambda$ a verifier V^*_λ that sends $(\mathsf{arg}_1, \mathsf{ct})$ as its first message, obtains arg_2 from P, feeds it to $\bar{\mathsf{V}}^*$ and outputs whatever $\bar{\mathsf{V}}^*$ does.

- 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.

Let $\mathsf{V}^* = \{\mathsf{V}^*_\lambda\}_\lambda$ be a (w.l.o.g deterministic) polynomial-size verifier. Assume toward contradiction that there exist an infinite set X of $x \in \mathcal{L} \cap \{0,1\}^\lambda$ and $w \in \mathcal{R}_\mathcal{L}(x)$ such that for some noticeable $\delta(\lambda)$

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

We argue that for any x such as above the verifier's message arg_1 is in the support of the honest verifier. Indeed, if it is not, then the reduction R first emulates a prover abort just as in the real system, where the prover detects the witness encryption does decrypt to randomness that explains arg_1. Thus,

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

Thus, we can construct an explainable verifier $\mathsf{V}^*_{x x\in X}$ that on input x behaves like V^*, and on any other input sends an abort message $\perp$. This verifier is explainable and fails the witness-finding reduction. □

Remark 5.2. The transformation given by Protocol 5 preserves public verifiability. Indeed, verification is the same as in the original protocol.

References

[AH91] William Aiello and Johan Håstad. Statistical zero-knowledge languages can be recognized in two rounds. *J. Comput. Syst. Sci.*, 42(3):327–345, 1991.

[AIR01] William Aiello, Yuval Ishai, and Omer Reingold. Priced oblivious transfer: How to sell digital goods. In *EUROCRYPT*, volume 2045 of *Lecture Notes in Computer Science*, pages 119–135. Springer, 2001.

[AJ17] Prabhanjan Ananth and Abhishek Jain. On secure two-party computation in three rounds. In *Theory of Cryptography - 15th International Conference, TCC 2017, Baltimore, MD, USA, November 12-15, 2017, Proceedings, Part I*, pages 612–644, 2017.

[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.

[CGGM00] Ran Canetti, Oded Goldreich, Shafi Goldwasser, and Silvio Micali. Resettable zero-knowledge (extended abstract). In *STOC*, pages 235–244. ACM, 2000.

[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.

[CS02] Ronald Cramer and Victor Shoup. Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption. In *Advances in Cryptology - EUROCRYPT 2002, International Conference on the Theory and Applications of Cryptographic Techniques, Amsterdam, The Netherlands, April 28 - May 2, 2002, Proceedings*, pages 45–64, 2002.

[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.

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

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

[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.

[Nao91] Moni Naor. Bit commitment using pseudorandomness. *J. Cryptology*, 4(2):151–158, 1991.

[NP01] Moni Naor and Benny Pinkas. Efficient oblivious transfer protocols. In S. Rao Kosaraju, editor, *Proceedings of the Twelfth Annual Symposium on Discrete Algorithms, January 7-9, 2001, Washington, DC, USA.*, pages 448–457. ACM/SIAM, 2001.

[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.

[PRS02] Manoj Prabhakaran, Alon Rosen, and Amit Sahai. Concurrent zero knowledge with logarithmic round-complexity. In *43rd Symposium on Foundations of Computer Science (FOCS 2002), 16-19 November 2002, Vancouver, BC, Canada, Proceedings*, pages 366–375, 2002.

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

[SV97] Amit Sahai and Salil P. Vadhan. A complete promise problem for statistical zero-knowledge. In *38th Annual Symposium on Foundations of Computer Science, FOCS '97, Miami Beach, Florida, USA, October 19-22, 1997*, pages 448–457, 1997.

[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.

A Relaxed Random Self-Reducible Public-Key Encryption from LWE

In this appendix, we describe a relaxed random self-reducible public-key based on LWE.

Lattice preliminaries. In what follows, q is a prime and α, β, m, n are such that $\alpha 2^{n^{0.1}} < \beta < \frac{q}{8m}$ and $n < \frac{m}{3\log q} < n^{O(1)}$. We denote by $\overline{\Psi}_{m,\sigma}$ the truncated discrete Gaussian distribution on $\mathbb{Z}^m$ with parameter σ, where any sample x such that $\|x\| > \sqrt{m}\sigma$ is replaced by 0. Following common habit, we will identify the security parameter (denoted by λ in the rest of the paper) with the lattice dimension n.

The scheme:

- $\mathsf{RSR.Gen}(1^n)$: outputs a public key $\mathbf{A} \in \mathbb{Z}_q^{m\times n}$ and secret key $\mathbf{t} \in \{0,1\}^{m-1}$ such that $\mathbf{A}$ is statistically close to uniform and $(1|\mathbf{t}^T)\mathbf{A} = 0$.
- $\mathsf{RSR.Enc}_{\mathbf{A}}(b)$: samples $\mathbf{s} \leftarrow \mathbb{Z}_q^n$, $\mathbf{e} \leftarrow \overline{\Psi}_{m,\alpha}$ and outputs $\mathbf{c} = \mathbf{A}\mathbf{s} + \mathbf{e} + b\left(\frac{q}{2}|0^{m-1}\right)$.
- $\mathsf{RSR.Dec}_{\mathbf{t}}(\mathbf{c})$: outputs 0 if $|\langle \mathbf{c}, (1|\mathbf{t}^T)\rangle| \leq q/4$, and 1 otherwise.
- $\mathsf{RSR}.\widetilde{\mathsf{Enc}}_{\mathbf{A}}(b)$: similar to $\mathsf{RSR.Enc}$, only that $\mathbf{e} \leftarrow \overline{\Psi}_{m,\beta}$.
- $\mathsf{RSR}.\widetilde{\mathsf{Dec}}^{\mathsf{D}}(\mathbf{c}, \mathbf{A}, 1^k)$:
  - Let $\widetilde{\mathbf{c}}$ be the distribution given by sampling $\mathbf{s}' \leftarrow \mathbb{Z}_q^n$, $\mathbf{e}' \leftarrow \overline{\Psi}_{m,\beta}$, and outputting $\mathbf{c} + \mathbf{A}\mathbf{s}' + \mathbf{e}'$.
  - Compute estimations:
    - $\tilde{\rho}$ of $\rho := \mathbb{E}\mathsf{D}(\widetilde{\mathbf{c}})$,
    - $\tilde{\rho}_0$ of $\rho_0 := \mathbb{E}\mathsf{D}(\mathsf{RSR}.\widetilde{\mathsf{Enc}}_{\mathbf{A}}(0))$,
    - $\tilde{\rho}_1$ of $\rho_1 := \mathbb{E}\mathsf{D}(\mathsf{RSR}.\widetilde{\mathsf{Enc}}_{\mathbf{A}}(1))$,

    using $k^2 \cdot n$ samples for each.
  - Output 0 if $|\rho - \rho_0| \leq |\rho - \rho_1|$ and 1 otherwise.

Claim A.1. *Assuming $LWE_{n,q,\alpha}$, the scheme is a relaxed RSR encryption.*

Proof sketch. We prove that the scheme satisfies the properties required in Definition 2.13.

Correctness: for any $b \in \{0,1\}, n \in \mathbb{N}$,

$$\mathsf{RSR.Dec}_{\mathbf{t}}(\mathsf{RSR.Enc}_{\mathbf{A}}(b)) = \left\langle \mathbf{A}\mathbf{s} + \mathbf{e} + b\left(\frac{q}{2} \;\middle|\; 0^{m-1}\right), (1|\mathbf{t}^T)\right\rangle = \\ 0 + \langle \mathbf{e}, (1|\mathbf{t}^T)\rangle + bq/2 \ .$$

Correctness then follows from the fact that

$$|\langle \mathbf{e}, (1|\mathbf{t}^T)\rangle| \leq \|\mathbf{e}\| \cdot \|(1|\mathbf{t}^T)\| \leq \alpha\sqrt{m} \cdot \sqrt{m} \leq q/8 \ .$$

For $\mathsf{RSR}.\widetilde{\mathsf{Enc}}_{\mathbf{A}}(b)$, correctness is shown similarly, where the only exception is that $\|\mathbf{e}\| \leq \beta\sqrt{m}$. Still,

$$|\langle \mathbf{e}, (1|\mathbf{t}^T)\rangle| \leq \beta m \leq q/8 \ \ .$$

Indistinguishability: by the $\mathrm{LWE}_{n,q,\alpha}$ assumption, for any polynomial-size distinguisher $\mathsf{D} = \{\mathsf{D}_n\}_{n\in\mathbb{N}}$, there exists a negligible μ such that for any security parameter $n \in \mathbb{N}$ and any $b \in \{0,1\}$,

$$\mathbf{A}, \mathsf{RSR}.\mathsf{Enc}_{\mathbf{A}}(b) = \mathbf{A}, \mathbf{A}\mathbf{s} + \mathbf{e} + b\left(\frac{q}{2} \;\middle|\; 0^{m-1}\right) \approx_{\mathsf{D}_n,\mu} \mathbf{A}, \mathbf{u} \ \ ,$$

where $\mathbf{A} \leftarrow \mathbb{Z}_q^{m\times n}, \mathbf{s} \leftarrow \mathbb{Z}_q^n, \mathbf{u} \leftarrow \mathbb{Z}_q^m$, and $\mathbf{e} \leftarrow \overline{\Psi}_{n,\alpha}$.

Random self-reduction: fix any $\mathbf{A}$ and distinguisher D such that $|\rho_0 - \rho_1| \geq \varepsilon$, and fix any ciphertext $\mathbf{c} = \mathbf{A}\mathbf{s} + \mathbf{e} + b\left(\frac{q}{2} \;\middle|\; 0^{m-1}\right) \in \mathsf{RSR}.\mathsf{Enc}_{\mathbf{A}}(b)$.

For this, we rely on *noise flooding* [Gen09a]: for $\mathbf{s}' \leftarrow \mathbb{Z}_q^n, \mathbf{e}' \leftarrow \overline{\Psi}_{m,\beta}$

$$\widetilde{\mathbf{c}} = \mathbf{c} + \mathbf{A}\mathbf{s}' + \mathbf{e}' = \mathbf{A}(\mathbf{s} + \mathbf{s}') + (\mathbf{e} + b\left(\frac{q}{2} \;\middle|\; 0^{m-1}\right) + \mathbf{e}')$$

is statistically close to a fresh sample

$$\mathsf{RSR}.\widetilde{\mathsf{Enc}}_{\mathbf{A}}(b) = \mathbf{A}\mathbf{s}' + \mathbf{e}' \ \ .$$

Our random self reduction process now succeeds by standard concentration bounds.

□