Proofs of Ignorance and Applications to 2-Message Witness Hiding

Apoorvaa Deshpande* Yael Kalai†

September 23, 2018

Abstract

We consider the following paradoxical question: *Can one prove lack of knowledge?* We define the notion of *Proofs of Ignorance*, construct such proofs, and use these proofs to construct a 2-message witness hiding protocol for all of NP.

More specifically, we define a proof of ignorance (PoI) with respect to any language $L \in \mathsf{NP}$ and distribution $\mathcal{D}$ over instances in L. Loosely speaking, such a proof system allows a prover to generate an instance x according to $\mathcal{D}$ along with a proof that she does not know a witness corresponding to x. We construct construct a PoI protocol for any random self-reducible NP language L that is hard on average. Our PoI protocol is non-interactive assuming the existence of a common reference string.

We use such a PoI protocol to construct a *2-message witness hiding protocol for* NP with adaptive soundness. Constructing a 2-message WH protocol for all of NP has been a long standing open problem. We construct our witness hiding protocol using the following ingredients (where T is any super-polynomial function in the security parameter):

1. T-secure PoI protocol,
2. T-secure non-interactive witness indistinguishable (NIWI) proofs,[1]
3. T-secure rerandomizable encryption with strong KDM security,[2]

where the first two ingredients can be constructed based on the T-security of DLIN.

At the heart of our witness-hiding proof is a new non-black-box technique. As opposed to previous works, we do not apply an efficiently computable function to the code of the cheating verifier, rather we resort to a form of case analysis and show that the prover's message can be simulated in both cases, without knowing in which case we reside.

1 Introduction

Cryptography has always challenged the limits of what we believe is possible. With the elegant work of zero knowledge proofs [GMR89], it is possible to prove that a statement is true without revealing anything except its validity. Furthermore, with zero knowledge proofs-of-knowledge [FFS88, GMR89] it is also possible to prove *knowledge* of some secret without revealing anything about the secret.

In this work, we try to answer the following paradoxical question:

Can one prove lack of knowledge?

Intuitively, it seems impossible, since one can always *pretend* to be ignorant. We explore the settings in which one could give a convincing *proof of ignorance*. As a thought experiment, suppose that Alice

*Brown University, email: acdeshpa@cs.brown.edu. Part of this work was done at Microsoft Research.

†Microsoft Research, email: yael@microsoft.com.

[1] By NIWI we mean a one message WI proofs *without* a CRS.

[2] By strong KDM security we mean that for any *possibly inefficient* function f, it is computationally hard to distinguish between $\mathsf{Enc}(f(\mathsf{pk}))$ and $\mathsf{Enc}(0)$.

holds a locked box and wants to convince Bob that she *does not know* the contents inside the box. In general, Bob has no reason to believe Alice unless Alice provides some evidence on how she got the box in the first place. Suppose Bob trusts that Charlie gives locked boxes without revealing its contents and suppose that Alice is able to *prove* that she got the box from Charlie, then Bob might be convinced of Alice's assertion.

In the context of NP languages, a proof of ignorance (PoI) for an instance x being in an NP language L should convince the verifier that indeed x is in L and yet that the prover does not know a witness corresponding to x. The main question that we need to tackle is *where does the instance x come from?* Note that if the verifier generates a random instance x for the prover, and the language is hard on average, then a PoI is not needed. Moreover, if the prover and verifier generate x together through a two-party computation protocol, then again a PoI is not needed.

We are interested in the setting where a prover can generate an instance x along with a corresponding proof of ignorance *on her own*. For example, suppose there is a way for Alice to sample an instance x through a "provably random process". Then Alice can convince a verifier that she does not know a witness for x (assuming the language is hard on average). In some sense, a proof of ignorance is a proof that the instance x has been generated correctly with "good" randomness.

1.1 Proofs of Ignorance

We define proofs of ignorance (PoI) for any NP language L with respect to a distribution $\mathcal{D}$ over instances in L. We consider the setting where the prover generates an instance x on her own according to $\mathcal{D}$, and gives a proof that she generated x without knowing a corresponding witness.

Note that in the *random oracle model* [BR93], it is easy to construct a PoI for any $L \in \mathsf{NP}$ w.r.t. any distribution $\mathcal{D}$ over instances in L for which given a random u it is hard to find a witness corresponding to $x = \mathcal{D}(u)$. To generate an instance x distributed according to $\mathcal{D}$ together with a PoI do the following: Choose r at random, compute $u = H(r)$, and set $x = \mathcal{D}(u)$ to be the instance and r to be the PoI. Since H is modeled as a random oracle, knowledge of r (even if r is adversarially chosen) convinces the verifier that the prover uses a random u to generate x and thus does not know a witness corresponding to x, assuming our hardness assumption on L w.r.t. $\mathcal{D}$.

A PoI is impossible to achieve in the standard model without any interaction, since the prover can simply hardwire an instance-witness pair together with a PoI. We show that this is possible (for some NP languages) if the verifier sends a single message, in the form of a *common reference string* (CRS).

1.1.1 Defining Proofs of Ignorance

We define proofs of ignorance (PoI) with respect to a language $L \in \mathsf{NP}$ and a distribution $\mathcal{D}$ over instances in L. A PoI proof system for $(L, \mathcal{D})$, consists of a triplet of PPT algorithms $(\mathsf{Setup}, \mathsf{Gen}, \mathsf{Verify})$, such that Setup generates a common reference string (CRS). Any party can use the algorithm Gen, together with the CRS, to sample an instance x together with a PoI π, such that x is distributed according to $\mathcal{D}$.

The soundness guarantee we want is that if the verification algorithm Verify on input (CRS, x, π) outputs 1 then the prover who generated (x, π) does not know a witness corresponding to x. This is formalized by defining a computationally indistinguishable common reference string CRS' such that *for any* (x, π) if the algorithm Verify accepts (x, π) with respect to CRS', then it must be the case that $x \notin L$. Our intuition here is that since given CRS' one cannot generate a valid PoI together with a witness (since a witness does not exit), and since CRS' is computationally indistinguishable from CRS, it follows that given CRS one cannot generate a valid PoI together with a witness. We refer the reader to Definition 16 for the formal definition.

We also provide an alternative definition, which we refer to as *Trapdoor PoI*. In this definition, the CRS is generated together with a trapdoor td such that given (x, π, td) where π is a valid PoI with respect to x, it is easy to compute a valid witness w corresponding to x. The soundness guarantee is that given only the CRS (without the trapdoor), it is computationally infeasible for an adversary to output (x, π, w)

such that π is a valid PoI for x and w is a witness for x. We refer the reader to Definition 17 for the formal definition. Jumping ahead, we note that we use a trapdoor PoI in our 2-message witness hiding protocol.

1.1.2 Constructing Proofs of Ignorance

Let us start by giving a proof of ignorance protocol for the DDH language L_{DDH}. This language consists of elements of the form $x = (g^y, g^z, g^{yz})$, where g is a fixed generator of a primed order group $\mathbb{G}$, and where the corresponding witness is $w = (y, z)$. We construct a proof of ignorance for L_{DDH} with respect to the uniform distribution $\mathcal{U}_{\mathsf{DDH}}$, which samples an element in L_{DDH} by choosing at random $y, z \leftarrow \mathbb{Z}_p$ where $p = |\mathbb{G}|$ and outputting $x = (g^y, g^z, g^{yz})$.

The CRS of our PoI protocol is simply a random element in L_{DDH}; namely, $\mathsf{CRS} = (g^y, g^z, g^{yz})$ for randomly chosen $y, z \leftarrow \mathbb{Z}_p$. To generate a random element in L_{DDH} together with a PoI, simply choose at random $r, s \leftarrow \mathbb{Z}_p$, and let $x = (g^{yr}, g^{zs}, g^{yzrs})$ and the PoI be $\pi = (r, s)$. To check the validity of π simply check that indeed $x = \left((g^y)^r, (g^z)^s, (g^{yz})^{rs}\right)$. It is easy to see that by the discrete log assumption, if a prover given CRS generates an accepting (x, π) then the prover does not know a valid witness corresponding to x.

What enabled us to construct a proof of ignorance protocol for L_{DDH} is the fact that L_{DDH} is a random self-reducible language. More generally, we construct a proof of ignorance protocol for *any random self-reducible language*, where a language $L \in \mathsf{NP}$ is said to be random self-reducible if there exists a PPT algorithm f that converts any $x \in L$ into a uniformly distributed $x' = f(x, r) \in L$.

Theorem 1 (Informal). *If a language L is hard on average and is random self-reducible with respect to a distribution $\mathcal{D} = \{\mathcal{D}_\lambda\}_{\lambda \in \mathbb{N}}$ then there exists a proof of ignorance protocol for $(L, \mathcal{D})$.*

We also construct a trapdoor PoI (which is the primitive we use in our 2-message witness hiding protocol) for any random self-reducible language that is *witness preserving*. Loosely speaking, we say that a random self-reducible language $L \in \mathsf{NP}$ is witness preserving, if given a valid witness w for x, and given x' and r such that $x' = f(x, r)$, one can efficiently compute a valid witness for w' for x', and similarly, given x, x', r, w' such that $x' = f(x, r)$ and such that w' is a valid witness for x', one can efficiently compute a valid witness w for x. It is easy to see that L_{DDH} is a witness preserving random self-reducible language with respect to the uniform distribution $\mathcal{U}_{\mathsf{DDH}}$.

Theorem 2 (Informal). *If a language L is hard-to-extract with respect to a distribution $\mathcal{D}$,[3] and is witness-preserving random self-reducible with respect to $\mathcal{D}$, then there exists trapdoor proof of ignorance protocol for $(L, \mathcal{D})$.*

We prove these theorems in Section 5.

1.2 Witness Hiding from Proofs of Ignorance

Witness hiding proofs were introduced by Feige and Shamir [FS90]. Intuitively, an interactive proof for an NP language L is said to be *witness hiding* if participating in the protocol does not help the verifier find a witness corresponding to the underlying instance. Witness hiding is a natural weakening of the security requirement of zero-knowledge, and can replace zero knowledge (ZK) in several applications.

Despite the fact that witness-hiding is a weaker requirement than ZK, almost all our candidate constructions of witness hiding protocols for NP are themselves zero-knowledge (or weak zero-knowledge). In particular, it is known that there do not exist 2-message (weak) zero-knowledge protocols for NP [GO94], and indeed constructing a 2-message witness hiding protocol for NP remained an important open problem.

In this work, we construct a 2-message witness hiding protocol for NP, using the following ingredients. Fix any super-polynomial function $T = \lambda^{\omega(1)}$, where λ is the security parameter. The ingredients are:

[3] Namely, for every poly-size adversary$\mathcal{A}$, the probability that $\mathcal{A}$ outputs the witness for an instance drawn from $\mathcal{D}$ is negligible.

1. T-secure trapdoor proof of ignorance protocol.

2. T-secure non-interactive witness indistinguishable (NIWI) proofs.[4]

3. T-secure rerandomizable encryption with strong KDM security, where by strong KDM security we mean that for any *possibly inefficient* function f, it is computationally hard to distinguish between $\mathsf{Enc}(f(\mathsf{pk}))$ and $\mathsf{Enc}(0)$.

Remark 1. *The definition of strong KDM security was recently given in [CCRR18]. However, they require that* $\mathsf{Enc}(f(\mathsf{pk}))$ *and* $\mathsf{Enc}(0)$ *are indistinguishable with respect to an* exponential size *adversary, whereas we only require that they are indistinguishable with respect to a polynomial size adversary.*

Theorem 3 (Informal)**.** *Assuming the existence of the ingredients above, there exists a 2-message witness hiding protocol with adaptive soundness for* NP*.*

We note that Groth, Ostrovsky and Sahai [GOS06] construct a T-secure NIWI from the T security of the DLIN assumption. In this work, we construct a T-secure trapdoor proof-of-ignorance protocol under the same assumption. Thus, we obtain the following corollary.

Corollary 1. *There exists a 2-message witness hiding protocol with adaptive soundness for* NP *assuming that DLIN is T-secure and assuming a T-secure rerandomizable encryption with strong KDM security.*

Remark 2. *We have several constructions of T-secure rerandomizable encrypton schemes from standard assumptions, such as the T-security of DDH, or the T-security of quadratic residuosity. We do not know how to prove that these schemes are strong KDM secure under standard assumptions, but we do not have any evidence that they are not.*

1.2.1 Related Work on Witness Hiding Protocols

3-Message Protocols Most witness hiding protocols in the literature are also zero-knowledge. It is known that 3-message ZK protocols with black-box simulation do not exist [GK96]. Several 3-message ZK protocols with non-black-box simulation were constructed: Most known constructions are based on auxiliary-input knowledge assumptions [BP04, HT98, CD09, BCC$^+$17],[5] and very recently Bitanski *et. al.* [BKP18] gave a construction based on multi-collision-resistant hash functions. 3-message ZK protocols have also been constructed under standard assumptions in restricted adversarial models, where either the verifier or the prover is assumed to be uniform [BCPR16, BBK$^+$16].

The only example of a 3-message WH protocol which is not ZK, is by Bitansky and Paneth [BP12]. They rely on the assumption that there exist auxiliary input point-function obfuscators that satisfy a distributive requirement.[6]

2-Message Protocols It is well known that 2-message zero-knowledge protocols do not exist [GO94]. Indeed, constructing a 2-message witness hiding protocol for all of NP remained an elusive task. However, significant progress on this question has been made.

Loosely speaking, Feige and Shamir [FS90] observed that if a language has two *independent* witnesses then witness indistinguishability implies witness hiding. Importantly, constructing a 2-message witness indistinguishable (WI) protocol, and even a non-interactive WI protocol, is known for all of NP under various (standard) assumptions [DN00, BOV05, GOS06]. Pass [Pas03] used this observation to construct a 2-message ZK protocol with quasi-polynomial simulation for all of NP. Roughly speaking, his protocol follows the following blueprint: The verifier sends $y = f(r)$ where r is a random string and f is one-way

[4] By NIWI we mean a one message WI proofs *without* a CRS.

[5] These assumptions are believed to be false assuming that indistinguishability obfuscation exists [BCPR16].

[6] This assumption is believed to be false assuming that Virtual Grey Box obfuscation exists [BCKP17].

function that is invertible in quasi-polynomial time, and such that every element in the range has a preimage. The prover then sends a commitment c and gives a WI proof that $x \in L$ or that c commits to r' such that $f(r') = y$. Simulation works by inverting y in quasi-polynomial time and using that as a witness in the WI proof.

We note that this protocol is WH for super-polynomial hard languages. More generally, any protocol that is ZK with T-time simulation is WH for T-hard languages. The reason is that if (by contradiction) the resulting protocol is not WH, then one can find a witness in time roughly T, by simulating the prover (in time T) and then extracting a witness from the simulated transcript (in polynomial time). This contradicts the T-hardness of the language.

In this work, we construct a WH protocol for *all of* NP. We follow the blue-print of Pass, where we use our proof of ignorance protocol to construct an independent witness, and as a result avoid putting any restrictions on the hardness of L. We refer to Section 2 for details.

Other WH Protocols Jain *et. al.* [JKKR17] construct 2-message WH protocols (and distributional ZK) under standard assumptions, in the *delayed input* setting, where the instance is only determined by the prover in the last round.

There have been several works on witness hiding protocols for languages where each instance has a unique witness. Haitner, Rosen and Shaltiel [HRS09] showed that such languages do not have constant round public-coin witness hiding protocols which are based on standard assumptions via some restricted types of black-box reductions. Deng et al. [DSYC17] showed that for any such language L, and for any distribution $\mathcal{D}$ over L that has an indistinguishable counterpart distribution over a relation with multiple witnesses, it holds that any witness indistinguishable protocol is witness hiding with respect to $\mathcal{D}$. Bellare and Palacio [BP02] showed that the Schnorr and Guillou-Quisquater 3-message identification protocols are witness hiding under the assumptions of one-more Discrete Log and one-more RSA.

2 Technical Overview: Witness Hiding Arguments

The main technical contribution of this work is a 2-message witness hiding protocol from proofs of ignorance. This protocol, as well as its analysis, contain a novel non-black-box technique, which is of conceptual interest. Starting with the seminal work of Barak [BGI$^+$01], most non-black-box techniques use the code of the cheating verifier V^* in an "efficient manner" (eg., the simulator commits to the code of V^* and proves that this code satisfies a desired property). To prove that our protocol is witness hiding, we do not use the code of V^* in an efficient manner; rather, we resort to a form of case analysis. We argue that either it is possible to efficiently generate some trapdoor, in which case we can simulate the prover's message in a certain way, or the trapdoor cannot be generated efficiently, in which case we can simulate the prover's message in a different way. However, we do not know in which case we reside. This is what distinguishes our WH proof from a ZK proof.

In what follows we give an overview of our construction and proof of security. At a very high-level, we follow the approach of Pass [Pas03]. Our starting point is the observation of [FS90] that if a language L has two independent witnesses then a WI proof for L is also WH. We use this observation to construct a 2-message WH protocol for any language $L \in \mathsf{NP}$ and use our proof of ignorance protocol to generate an additional independent witness (corresponding to an independent instance).

The basic blueprint of our protocol is the following: The prover will generate an independent instance x' and prove that either $x' \in L$ or $x \in L$, using a 2-message WI proof. This 2-message protocol is definitely witness hiding, but it is not sound, since the prover can cheat and prove that $x \in L$ (even though this is false) by generating $x' \in L$ and using a witness w' for x' to convince the verifier. We overcome this obstacle by using a proof-of-ignorance; the prover will send x' together with a proof of ignorance, and will then prove that either $x \in L$ or $x' \in L$.

The problem here is that we do not have proof of ignorance (PoI) protocol for all of NP. This problem can be bypassed quite easily by choosing some language $L' \in \mathsf{NP}$ that has a PoI protocol, and now the prover will generate $x' \in L'$ together with a PoI, and will add a WI proof that $x \in L$ or $x' \in L'$.

Attempt 1. Our first attempt at constructing a 2-message WH protocol is the following:

- **Verifier's message:** Verifier samples CRS corresponding to the PoI proof system for L', and sends it to the prover.
- **Prover's message:** The prover samples $x' \in L'$ with a proof of ignorance π', and send (x', π') along with a WI proof for the following language:

$$L_{\mathsf{WI}} = \{(x, x') \mid \exists\, w \text{ such that } (x, w) \in R_L \ \vee\ (x', w) \in R_{L'}\}$$

Intuitively, this 2-message protocol seems to be sound, since if $x \notin L$ and if the prover generates a valid PoI for x' then he does not know a witness to x (since one does not exist) nor to x', and thus cannot cheat. The actual proof is quite subtle, and in particular requires using a trapdoor PoI, and relying on super-polynomial hardness assumptions. Subsequently, we elaborate on these subtleties.

However, a more serious problem here is that adding the PoI seems to damage the WH property. For example, the (cheating) verifier can generate CRS maliciously in a way that if π' is a valid PoI for x' then it must be the case that $x' \notin L'$, and thus we do not have two independent witnesses, and hence the WI property may not protect us at all.

We fix this problem by having the verifier not only send the CRS for the PoI, but also prove that it is "well formed". Namely, he proves that there exists randomness that "explains" this CRS. There are two issues with adding this proof of correctness: First, it seems like we need a PoI with the strong property that *for every* well formed CRS, if the prover is honest then he generates a randomly distributed $x' \in L'$, *independent* of CRS, whereas our PoI have this property only for an honestly generated CRS. Here again, the trapdoor PoI comes to the rescue (as we explain in more detail below).

The other issue is that for soundness it is crucial that the prover does not learn sensitive information about the CRS (in particular, how it was generated). Thus, it seems that to maintain soundness, the verifier will need to use a zero-knowledge proof, or at least a witness-hiding proof, which brings us back to where we started from in the first place!

We solve the latter problem using the same blueprint that we started with. Rather than sending a single CRS, the verifier will send two independent copies CRS_0 and CRS_1 and will give a WI proof that at least one of them is well formed. Since the verifier sends the first message we need to rely on a *non-interactive WI* (NIWI) proof. The prover will then send (x'_0, π'_0) corresponding to CRS_0 and (x'_1, π'_1) corresponding to CRS_1, and will give a NIWI proof that either $x \in L$ or $x'_0 \in L'$ or $x'_1 \in L'$.

Attempt 2. Our second attempt at constructing a 2-message WH protocol is the following:

- **Verifier's message:** Verifier independently samples CRS_0 and CRS_1 corresponding to the PoI proof system for L', and generates a NIWI proof π_{NIWI} that at least one of them is well-formed. He sends $(\mathsf{CRS}_0, \mathsf{CRS}_1, \pi_{\mathsf{NIWI}})$ to the prover.
- **Prover's message:** The prover first checks that the NIWI proof π_{NIWI} is valid, and if not aborts. Otherwise, for every $b \in \{0, 1\}$, the prover samples $x'_b \in L'$ with its proof of ignorance π'_b, and sends (x'_0, π'_0), (x'_1, π'_1), along with a NIWI proof that $(x \in L)$ or $(x'_0 \in L')$ or $(x'_1 \in L')$.

Intuitively, the soundness of this protocol follows from the fact that if $x \notin L$ then the only way to cheat is by using either a witness for x'_0 or a witness for x'_1, and the PoI guarantees that a cheating prover does not know such a witness. However, to argue this formally, a NIWI does not suffice and we need a NIWI *proof of knowledge*. This is the case since the PoI only guarantees that it is hard to find a witness, and not that a witness does not exist. We achieve this proof-of-knowledge property by resorting to complexity leveraging. Namely, the prover will also send a (statistically binding) commitment c and will prove that either $x \in L$ or that there exists $b \in \{0,1\}$ such that c is a commitment to a valid witness of x'_b. Suppose this commitment can be broken in time T, then we can *extract* from the cheating prover a witness w'_b to x'_b, and argue that this breaks the soundness of the underlying PoI, (which asserts that given CRS_b it is hard to generate (x'_b, π'_b, w'_b)). However, to argue this formally, one needs to assume T-security of the PoI protocol, and T-security of the WI protocol. Then we can prove that the following protocol is indeed sound.

Attempt 3. Our third attempt at constructing a 2-message WH protocol is the following:

- **Verifier's message:** Verifier samples independently CRS_0 and CRS_1 corresponding to the PoI proof system for L', and generates a NIWI proof π_{NIWI} that at least one of them is well-formed. He sends $(\mathsf{CRS}_0, \mathsf{CRS}_1, \pi_{\mathsf{NIWI}})$ to the prover.
- **Prover's message:** The prover does the following:
  1. Check that the NIWI proof π_{NIWI} is valid, and if not abort.
  2. For every $b \in \{0,1\}$, use CRS_b to sample $x'_b \in L'$ along with a proof of ignorance π'_b.
  3. Compute c which is a commitment to 0.
  4. Compute a NIWI proof π'_{NIWI} that $x \in L$ or that there exists $b \in \{0,1\}$ for which c is a commitment to a valid witness corresponding to x'_b.
- Send $\big((x'_0, \pi'_0), (x'_1, \pi'_1), c, \pi'_{\mathsf{NIWI}}\big)$.

We can formally argue that this protocol is sound. However, it is still not clear how to argue that it is WH. As mentioned before, the problem is that the CRS could still be maliciously chosen (albeit well-formed). Our first observation is that this protocol is WH against cheating verifiers who "know" a valid trapdoor td_b corresponding to CRS_b (for some $b \in \{0,1\}$). This is true because given td_b one can efficiently compute w'_b from (x'_b, π'_b), and thus simulate the NIWI proof of the prover efficiently.

Thus, restating the problem: What if the cheating verifier managed to construct a valid NIWI proof without knowing a valid trapdoor to CRS_0 or CRS_1? Again, this would have been solved with a NIWI proof-of-knowledge. However, for one-message NIWI PoK we would need complexity leveraging and the use of complexity leveraging here would result in a WH protocol only for T-hard languages.[7]

Our Non-Black-Box Technique: We overcome the hurdle described above by instructing the verifier to encrypt the trapdoors of each CRS and prove that one of these encryptions indeed encrypts a valid trapdoor. Namely, the verifier does the following: Sample two fresh and independent public keys $(\mathsf{pk}_0, \mathsf{pk}_1)$ corresponding to a semantically secure encryption scheme, and send $\{(\mathsf{CRS}_b, \mathsf{pk}_b, \mathsf{ct}_b)\}_{b\in\{0,1\}}$, where $\mathsf{ct}_b \leftarrow \mathsf{Enc}_{\mathsf{pk}_b}(\mathsf{td}_b)$, along with a NIWI proof that there exists $b \in \{0,1\}$ such that ct_b is an encryption of a valid td_b corresponding to CRS_b.

As explained above, we cannot afford to extract a trapdoor from the ciphertexts (since this may take super-polynomial time). Instead, we instruct the prover to give its proof of ignorance π'_b *encrypted* under

[7] This approach was used by Pass in [Pas03].

pk_b. This allows us to prove witness hiding using the following non-black-box approach. We distinguish between the following two cases:

- **Case 1:** The verifier "knows" a trapdoor corresponding to CRS_0 or CRS_1. In this case, one can efficiently simulate the prover's message using this trapdoor, and thus WH holds (as was argued before).
- **Case 2:** The verifier does not know a trapdoor to CRS_0 or to CRS_1. In this case, we argue that the verifier cannot distinguish between the case that the prover encrypts a valid PoI π'_b corresponding to x'_b, or encrypts 0, and thus again one can efficiently simulate the prover's message by encrypting 0, and generating x'_b together with a valid witness w'_b. To argue that indeed the verifier cannot distinguish between $\mathsf{Enc}_{\mathsf{pk}_b}(\pi'_b)$ and $\mathsf{Enc}_{\mathsf{pk}_b}(0)$ we rely on an encryption scheme that is rerandomizable. However, we need something stronger: Namely, we need to argue that the verifier cannot distinguish between $\mathsf{Enc}_{\mathsf{pk}_b}(\pi'_b)$ and $\mathsf{Enc}_{\mathsf{pk}_b}(0)$ given the NIWI proof π'_{NIWI}. To prove this we need the assumption that this encryption scheme is strong KDM secure. We defer the details to Section 6.

Attempt 4. Our fourth (and almost final) attempt is the following.

- **Verifier's message:** The verifier does the following:
  1. Sample independently two public keys pk_0 and pk_1 (corresponding to a rerandomizable T-secure strong KDM secure encryption scheme).
  2. Sample independently CRS_0 and CRS_1, together with corresponding trapdoors td_0 and td_1.
  3. Generate $\mathsf{ct}_0 \leftarrow \mathsf{Enc}_{\mathsf{pk}_0}(\mathsf{td}_0)$ and $\mathsf{ct}_1 \leftarrow \mathsf{Enc}_{\mathsf{pk}_1}(\mathsf{td}_1)$.
  4. Generate a NIWI proof π_{NIWI} that there exists $b \in \{0,1\}$ for which ct_b encrypts a valid trapdoor corresponding to CRS_b.

  Send $\Big((\mathsf{CRS}_0, pk_0, \mathsf{ct}_0), (\mathsf{CRS}_1, pk_1, \mathsf{ct}_1), \pi_{\mathsf{NIWI}}\Big)$ to the prover.
- **Prover's message:** The prover does the following:
  1. Check that the NIWI proof π_{NIWI} is valid, and if not abort.
  2. For every $b \in \{0,1\}$, use CRS_b to sample $x'_b \in L'$ along with a proof of ignorance π'_b.
  3. For every $b \in \{0,1\}$, generate $\mathsf{ct}'_b \leftarrow \mathsf{Enc}_{\mathsf{pk}_b}(\pi'_b)$.
  4. Compute c which is a commitment to 0.
  5. Compute a NIWI proof π'_{NIWI} that $x \in L$ or that there exists $b \in \{0,1\}$ for which c is a commitment to a valid witness corresponding to x'_b.
- Send $\Big((x'_0, \mathsf{ct}'_0), (x'_1, \mathsf{ct}'_1), c, \pi'_{\mathsf{NIWI}}\Big)$.

We can indeed argue that this protocol is WH, as argued above. However, to argue soundness recall that we need to extract from the cheating prover a tuple (x'_b, w'_b, π'_b). Previously, π'_b was given in the clear, and w'_b was extracted in time T from the commitment. However, now we also need to extract π'_b, which will take more time, since the encryption is T-secure. Instead, we instruct the prover to generate a commitment $c'_b = \mathsf{Com}(\pi'_b; r'_b)$, in addition to ct'_b, and compute a NIWI proof π'_{NIWI} that $x \in L$ or that there exists $b \in \{0,1\}$ for which c is a commitment to a valid witness corresponding to x'_b and ct'_b is an encryption to a pair (π'_b, r'_b) such that $c'_b = \mathsf{Com}(\pi'_b; r'_b)$.

The formal protocol and the proof can be found in Section 6.

3 Preliminaries

Notation: We denote the security parameter by λ. We use PPT to denote that an algorithm is probabilistic polynomial time. Suppose A is a probabilistic algorithm, then we denote by $y \leftarrow A(x)$ the event that y is generated by sampling randomness $r \overset{\$}{\leftarrow} \{0,1\}^*$ and setting $y = A(x; r)$.

We say that a function $\nu : \mathbb{N} \to \mathbb{N}$ is negligible (sometimes denoted by negl) if for every polynomial p there exists $\lambda_0 \in \mathbb{N}$ such that for all $\lambda > \lambda_0$, $\nu(\lambda) < 1/p(\lambda)$. For any language L, we denote by $L = \{L_\lambda\}_{\lambda \in \mathbb{N}}$ where $L_\lambda = L \cap \{0,1\}^\lambda$. We use the notation of $\{\mathcal{X}_\lambda\}_{\lambda \in \mathbb{N}} \approx_c \{\mathcal{Y}_\lambda\}_{\lambda \in \mathbb{N}}$, and $\{\mathcal{X}_\lambda\}_{\lambda \in \mathbb{N}} \approx_s \{\mathcal{Y}_\lambda\}_{\lambda \in \mathbb{N}}$, to denote that the distribution ensembles $\{\mathcal{X}_\lambda\}_{\lambda \in \mathbb{N}}$ and $\{Y_\lambda\}_{\lambda \in \mathbb{N}}$ are computationally and statistically indistinguishable, respectively.

3.1 Witness Hiding and Witness Indistinguishable Proofs

Definition 1 (Interactive proofs)**.** An **interactive proof** system for an NP language L, with a corresponding NP-relation R_L, is a protocol $\langle P, V \rangle$, between a PPT prover P and a PPT verifier V, at the end of which V outputs a bit b, and such that the following two properties are satisfied.

- **Completeness.** There exists a negligible function μ such that for every $\lambda \in \mathbb{N}$ and every $(x, w) \in R_L$,
$$\Pr[b \leftarrow \langle P(w), V \rangle(1^\lambda, x) \;:\; b = 1] \geq 1 - \mu(\lambda)$$
where the probability is over the random coin tosses of P and V.

- **Soundness.** For any cheating prover P^* there exists a negligible function μ such that for every $\lambda \in \mathbb{N}$ and every $x \notin L$,
$$\Pr[b \leftarrow \langle P^*, V \rangle(1^\lambda, x) \;:\; b = 1] \leq \mu(\lambda),$$
where the probability is over the random coin tosses of V.

  We say that $\langle P, V \rangle$ has **perfect soundness** if the soundness condition holds with $\mu = 0$ (for any P^*).

Definition 2 (Interactive Arguments)**.** An **interactive argument** system for a language L is a protocol $\langle P, V \rangle$ as in Definition 1, where the completeness property is as before, but the soundness property is relaxed as follows.

- **Computational Soundness.** For any poly-size cheating prover P^* there exists a negligible function μ, such that for every $\lambda \in \mathbb{N}$, and every $x \in \{0,1\}^{\mathsf{poly}(\lambda)} \setminus L$,
$$\Pr[b \leftarrow \langle P^*, V \rangle(1^\lambda, x) \;:\; b = 1] \leq \mu(\lambda)$$
where the probability is over the random coin tosses of V.

A 2-message argument system $\langle P, V \rangle$ for a language L is one that consists of only two messages, the first sent from the verifier to the prover and the second sent from the prover to the verifier. In a 2-message argument system, we often denote the verifier by $V = (V_1, V_2)$, where V_1 generates the message to be sent to the prover, and V_2 checks the validity of the proof given by the prover. If the message of the verifier V_1 does not depend on the instance x (and depends only on the security parameter), then we often denote $(\mathsf{pp}, \mathsf{st}) \leftarrow V_1(1^\lambda)$, where pp is the message sent to the prover and st is the secret state used by V_2 to verify the proof.[8]

[8] If $\mathsf{st} = \emptyset$ then such an argument system is said to be publicly verifiable.

Definition 3 (2-Message Arguments with Adaptive Soundness)**.** *A 2-message argument system $\langle P, V\rangle$ for an* NP *language L, where the message sent by the verifier is independent of the instance x, is said to have* **adaptive soundness** *if for any poly-size cheating prover P^* there exists a negligible function μ such that*

$$\Pr\left[(x, \mathsf{msg}_P) = P^*(1^\lambda, x, \mathsf{pp}) \text{ s.t. } \big(x \notin L\big) \ \wedge \ \big(V_2(1^\lambda, x, \mathsf{pp}, \mathsf{st}, \mathsf{msg}_P) = 1\big)\right] \leq \mu(\lambda)$$

where the probability is over $(\mathsf{pp}, \mathsf{st}) \leftarrow V_1(1^\lambda)$ and over the random coin tosses of V_2.[9]

Definition 4 (Witness Hiding [FS90])**.** *Let $L \in$* NP *and let R_L be the corresponding* NP *witness relation. Let $\mathcal{D} = \{\mathcal{D}_\lambda\}_{\lambda\in\mathbb{N}}$ be a* PPT *distribution ensemble over instances in R_L. An interactive proof system $\langle P, V\rangle$ for L is* **witness hiding** *with respect to $\mathcal{D}$ if the following holds:*
For every poly-size V^* there exists a negligible function μ such that for every $\lambda \in \mathbb{N}$,

$$\Pr_{(x,w)\leftarrow\mathcal{D}_\lambda}[\langle P(w), V^*\rangle(1^\lambda, x) \in R_L(x)] = \mu(\lambda)$$

Definition 5 (Non Interactive Witness Indistinguishable (NIWI) proofs)**.** Let $L \in \mathsf{NP}$ and let R_L be the corresponding NP relation. A pair of algorithms $(\mathsf{Prove}, \mathsf{Verify})$ is called a *non-interactive witness indistinguishable* (NIWI) [FS90, GOS06] proof system (or argument system) for L if it satisfies completeness and soundness as in Definition 1 (or Definition 2), and in addition it satisfies the following witness indistinguishability property:
For every poly-size adversary $\mathcal{A}$, there exists a negligible function ν such that for every $\lambda \in \mathbb{N}$, the probability that $b' = b$ in the following game is at most $1/2 + \nu(\lambda)$:

1. $(\mathsf{state}, x, w_0, w_1) = \mathcal{A}(1^\lambda)$
2. Choose $b \xleftarrow{\$} \{0, 1\}$. If $R_L(x, w_0) \neq 1$ or $R_L(x, w_1) \neq 1$ then output $\perp$. Else, set $\pi \leftarrow \mathsf{Prove}(1^\lambda, x, w_b)$.
3. $b' = \mathcal{A}(\mathsf{state}, \pi)$

Theorem 4 ([GOS06])**.** *There exists a perfectly sound NIWI proof system based on the DLIN assumption (defined below).*

Assumption 1 (Decisional Linear Assumption)**.** A bilinear group generation algorithm $\mathcal{G}$ on input the security parameter 1^λ outputs $(p, \mathbb{G}, \mathbb{G}_T, e, g)$, where $\mathbb{G}$ and $\mathbb{G}_T$ are groups of order p, $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ is a bilinear map, and g is a generator of $\mathbb{G}$. The Decisional Linear (DLIN) Assumption holds for a bilinear group generator $\mathcal{G}$ if the following distributions are computationally indistinguishable:

$$\{(p, \mathbb{G}, \mathbb{G}_T, e, g) \leftarrow \mathcal{G}(1^\lambda) \ ; \ (x, y) \xleftarrow{\$} \mathbb{Z}_p^* \ ; \ (r, s) \xleftarrow{\$} \mathbb{Z}_p \ : \ (p, \mathbb{G}, \mathbb{G}_T, e, g, g^x, g^y, g^{xr}, g^{ys}, g^{r+s})\}_{\lambda\in\mathbb{N}} \text{ and}$$

$$\{(p, \mathbb{G}, \mathbb{G}_T, e, g) \leftarrow \mathcal{G}(1^\lambda) \ ; \ (x, y) \xleftarrow{\$} \mathbb{Z}_p^* \ ; \ (r, s, d) \xleftarrow{\$} \mathbb{Z}_p \ : \ (p, \mathbb{G}, \mathbb{G}_T, e, g, g^x, g^y, g^{xr}, g^{ys}, g^d)\}_{\lambda\in\mathbb{N}}$$

Definition 6 (T-secure NIWI)**.** *A NIWI proof system (*$\mathsf{Prove}, \mathsf{Verify}$*) for L as in Definition 5 is said to be T-secure if witness indistinguishability holds for all adversaries of size* $\mathsf{poly}(T)$*.*

3.2 Encryption Schemes

Definition 7 (T-secure Encryption)**.** *A bit-wise encryption scheme* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *is said to be T-secure if semantic security holds for all adversaries of size* $\mathsf{poly}(T)$*. Namely, for every* $\mathsf{poly}(T)$*-size adversary $\mathcal{D}$, there exists a negligible function ν such that for every $\lambda \in \mathbb{N}$,*

$$\mathcal{P}[\mathcal{D}(\mathsf{pk}, \mathsf{ct}) = b] \leq 1/2 + \nu(T(\lambda)),$$

where the probability is over $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^\lambda)$*, $b \xleftarrow{\$} \{0, 1\}$ and* $\mathsf{ct} \leftarrow \mathsf{Enc}_{\mathsf{pk}}(b)$*.*

[9] Usually, in such argument systems V_2 is deterministic.

Remark 3. *For the simplicity of notation, we assume that* $|\mathsf{pk}| = \lambda$*, for every* pk *generated by* $\mathsf{Gen}(1^\lambda)$*, and we assume that* $\mathsf{Enc}_{\mathsf{pk}}$ *uses* λ *bits of randomness. We note that one can use a bit-wise encryption scheme to encrypt longer messages. Namely, for any* $k = \mathsf{poly}(\lambda)$ *one can encrypt any message* $m = (m_1, \ldots, m_k) \in \{0,1\}^k$ *as follows:*

$$\mathsf{Enc}_{\mathsf{pk}}(m) = \Big(\mathsf{Enc}_{\mathsf{pk}}(m_1), \ldots, \mathsf{Enc}_{\mathsf{pk}}(m_k)\Big).$$

The following lemma follows from a standard hybrid argument.

Lemma 1. *If a bit-wise encryption scheme* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *is* T*-secure then it is* T*-secure with respect to messages of length* $\mathsf{poly}(T(\lambda))$*. More concretely, for every adversary* $\mathcal{A}$ *of size* $\mathsf{poly}(T)$*, there exists a negligible function* ν *such that for every* $\lambda \in \mathbb{N}$*, the probability that* $b' = b$ *in the following game is at most* $1/2 + \nu(T(\lambda))$*:*

1. $(\mathsf{state}, m_0, m_1) \leftarrow \mathcal{A}(1^\lambda)$
2. *If* $|m_0| \neq |m_1|$ *then output* $\perp$.
3. *Else, compute* $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^\lambda)$*, choose* $b \xleftarrow{\$} \{0,1\}$*, and let* $\vec{\mathsf{ct}} = (\mathsf{ct}_1, \ldots, \mathsf{ct}_{|m_b|})$*, where* $\mathsf{ct}_i \leftarrow \mathsf{Enc}_{\mathsf{pk}}(m_{b,i})$ *for* $i \in [|m_b|]$.
4. $b' \leftarrow \mathcal{A}(\mathsf{state}, \mathsf{pk}, \mathsf{ct})$

Definition 8 (Rerandomizable Encryption)**.** *A* T*-secure bit-wise public-key encryption scheme* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$*, is said to be* **rerandomizable** *if there exists a* PPT *algorithm* Rand *that on input any pair* $(\mathsf{pk}, \mathsf{ct})$ *outputs a ciphertext* $\mathsf{ct}^{'}$ *with the following property: For every* $\mathsf{poly}(T)$*-size adversary* $\mathcal{A}$ *there exists a negligible function* ν *such that for every* $\lambda \in \mathbb{N}$*, the probability that* $b' = b$ *in the following game is at most* $1/2 + \nu(T(\lambda))$*:*

1. $\mathcal{A}(1^\lambda) = (\mathsf{state}, \mathsf{pk}, \mathsf{ct}, m, r)$.
2. *If* $\mathsf{ct} \neq \mathsf{Enc}_{pk}(m; r)$ *then output* $\perp$.
3. *Else, choose* $b \xleftarrow{\$} \{0,1\}$*. If* $b = 0$*, choose at random* $r' \leftarrow \{0,1\}^\lambda$ *and output* $\mathsf{ct}' = \mathsf{Enc}_{\mathsf{pk}}(m; r')$*. Else if* $b = 1$*, output* $\mathsf{ct}' = \mathsf{Rand}(\mathsf{pk}, \mathsf{ct})$.
4. $b' \leftarrow \mathcal{A}(\mathsf{state}, \mathsf{ct}')$

Lemma 2. *Let* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *be a poly-secure rerandomizable bit-wise public-key encryption scheme. For any poly-size distinguisher* $\mathcal{D}$ *and any polynomial* $q = \mathsf{poly}(\lambda)$*, there exists a non-uniform* PPT *distinguisher* $\mathcal{D}^*$*, such that for any* $\lambda \in \mathbb{N}$ *and any public key* pk *such that* $|\mathsf{pk}| = \lambda$*, the following holds: If*

$$\Pr[\mathcal{D}(\mathsf{pk}, \mathsf{Enc}_{\mathsf{pk}}(b)) = b] \geq \frac{1}{2} + \frac{1}{q(\lambda)} \tag{1}$$

where the probability is over $b \xleftarrow{\$} \{0,1\}$ *and over the randomness of* $\mathsf{Enc}_{\mathsf{pk}}$*, then for every* $b \in \{0,1\}$ *and every* $r \in \{0,1\}^\lambda$*,*

$$\Pr[\mathcal{D}^*(\mathsf{pk}, \mathsf{Enc}_{\mathsf{pk}}(b; r)) = b] \geq 1 - 2^{-\lambda} \tag{2}$$

where the probability is over the random coin tosses of $\mathcal{D}^*$*.*

Proof. Fix any poly-size distinguisher $\mathcal{D}$ and any polynomial q. We construct a non-uniform PPT distinguisher $\mathcal{D}^*$ as follows. Fix any pk that satisfies Equation (1) (w.r.t. $\mathcal{D}$ and q). Let $E_b(\mathsf{ct})$ denote the event that there exists randomness r such that $\mathsf{ct} = \mathsf{Enc}_{\mathsf{pk}}(b; r)$ and $\mathcal{D}(\mathsf{pk}, \mathsf{ct}) = b$. Denote by

$$p_b = \Pr[E_b(\mathsf{ct})],$$

where the probability is over $\mathsf{ct} \leftarrow \mathsf{Enc}_{\mathsf{pk}}(b)$. By Equation (1), we have that

$$\frac{p_0 + p_1}{2} \geq \frac{1}{2} + \frac{1}{q(\lambda)} \quad \text{or equivalently,} \quad \frac{p_0 + p_1 - 1}{2} \geq \frac{1}{q(\lambda)}$$

We now construct a non-uniform PPT distinguisher $\mathcal{D}^*$, that on input $(\mathsf{pk}, \mathsf{ct})$, where $\mathsf{ct} = \mathsf{Enc}_{\mathsf{pk}}(b^*; r)$, does the following:

1. Choose $N = \lambda \cdot 4q(\lambda)^2$. For each $i \in [N]$, compute $\mathsf{ct}'_i \leftarrow \mathsf{Rand}(\mathsf{pk}, \mathsf{ct})$ and $b_i = \mathcal{D}(\mathsf{pk}, \mathsf{ct}'_i)$.
2. Let $T = \frac{p_1 - p_0 + 1}{2} N$. If $\sum_{i=1}^{N} b_i < T$ then output 0, and else output 1.

By the definition of rerandomizable encryption (Definition 8), for every $b \in \{0,1\}$ there exists a negligible function ν_b such that

$$\Pr[\mathcal{D}(\mathsf{pk}, \mathsf{ct}'_i) = b^* \mid b^* = b] \geq p_b - \nu_b(\lambda) =: p'_b \tag{3}$$

where the probability is over $\mathsf{ct}'_i \leftarrow \mathsf{Rand}(\mathsf{pk}, \mathsf{ct})$.

Now we prove that Equation (2) holds for $b^* = 1$.

$$\begin{aligned}
&\Pr[\mathcal{D}^*(\mathsf{pk}, \mathsf{ct}) = b^* \mid b^* = 1] \\
&= \Pr\big[\sum_{i=1}^{N} b_i \geq \frac{p_1 - p_0 + 1}{2} N\big] \\
&= 1 - \Pr\big[\sum_{i=1}^{N} b_i < \frac{p_1 - p_0 + 1}{2} N\big] \\
&= 1 - \Pr\big[\sum_{i=1}^{N} b_i < N p'_1 - N\delta'\big] \text{ for } \delta' = \frac{p_0 + p_1 - 1}{2} - \nu_1(\lambda) \\
&\geq 1 - \Pr\big[\sum_{i=1}^{N} b_i < N p'_1 - N\delta\big] \text{ for } \delta = \frac{1}{2q(\lambda)} < \delta' \\
&\geq 1 - 2^{-\delta^2 N} \\
&\geq 1 - 2^{-\frac{N}{4q(\lambda)^2}} \\
&\geq 1 - 2^{-\lambda}
\end{aligned}$$

where the fifth equation follows from Chernoff bound.

A similar calculation shows that

$$\Pr[D^*(\mathsf{pk}, \mathsf{ct}) = b^* \mid b^* = 0] > 1 - 2^{-\lambda}.$$

Hence, we proved that Equation (2) holds for every $b^* \in \{0,1\}$ and every $r \in \{0,1\}^\lambda$. □

The following lemma follows from Lemma 2, together with a straightforward union bound.

Lemma 3. *Let* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *be a a poly-secure rerandomizable bit-wise public-key encryption scheme. For any poly-size distinguisher* $\mathcal{D}$ *and any polynomials* $q, k = \mathsf{poly}(\lambda)$*, there exists a non-uniform* PPT *distinguisher* $\mathcal{D}^*$ *such that for any* $\lambda \in \mathbb{N}$ *and any public key* pk *such that* $|\mathsf{pk}| = \lambda$*, the following holds: If*

$$\Pr[\mathcal{D}(\mathsf{pk}, \mathsf{Enc}_{\mathsf{pk}}(b)) = b] \geq \frac{1}{2} + \frac{1}{q(\lambda)},$$

where the probability is over $b \overset{\$}{\leftarrow} \{0,1\}$ *and over the randomness of* $\mathsf{Enc}_{\mathsf{pk}}$*, then for any message* $m = (m_1, \ldots, m_k) \in \{0,1\}^k$ *and any* $r_1, .., r_k \in \{0,1\}^\lambda$*,*

$$\Pr[\mathcal{D}^*(\mathsf{pk}, \vec{\mathsf{ct}}) = m] \geq 1 - \mathsf{negl}(\lambda)$$

where $\vec{\mathsf{ct}} = (\mathsf{ct}_1, \ldots, \mathsf{ct}_k)$ *and* $\mathsf{ct}_i = \mathsf{Enc}_{\mathsf{pk}}(m_i; r_i)$ *for* $i \in [k]$*, and where the probability is over the random coin tosses of* $\mathcal{D}^*$*.*

Definition 9 (Strong KDM Security)**.** *A semantically secure public-key encryption scheme* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *is said to be* **strong KDM secure** *if for every* PPT *distribution* $\mathcal{D}$ *used to (maliciously) sample public keys, if*

$$\big(\mathsf{pk}^*, \mathsf{Enc}_{\mathsf{pk}^*}(0)\big) \approx_c \big(\mathsf{pk}^*, \mathsf{Enc}_{\mathsf{pk}^*}(1)\big)$$

where $\mathsf{pk}^* \leftarrow \mathcal{D}(1^\lambda)$*, then for every (not necessarily efficient) function* f *such that* $f(\mathsf{pk}^*) \in \{0,1\}^{\mathsf{poly}(\lambda)}$*,*

$$\Big(\mathsf{pk}^*, \mathsf{Enc}_{\mathsf{pk}^*}(f(\mathsf{pk}^*))\Big) \approx_c \Big(\mathsf{pk}^*, \mathsf{Enc}_{\mathsf{pk}^*}\big(0^{\mathsf{poly}(\lambda)}\big)\Big)$$

where $\mathsf{pk}^* \leftarrow \mathcal{D}(1^\lambda)$*.*

3.3 Commitment Schemes

Definition 10 (Statistically Binding Commitments)**.** A statistically binding commitment scheme over a message space $\mathcal{M}$ consists of PPT algorithm Com which on input a message $m \in \mathcal{M}$ and randomness $r \in \{0,1\}^\lambda$ outputs a commitment $\mathsf{Com}(m; r)$.
We require the following properties from the commitment scheme:

Statistically Binding: For every $\lambda \in \mathbb{N}$, every $m_0, m_1 \in \mathcal{M}$ such that $m_0 \neq m_1$, and every $r_0, r_1 \in \{0,1\}^\lambda$,

$$\mathsf{Com}(m_0; r_0) \neq \mathsf{Com}(m_1; r_1)$$

Computationally Hiding: For every non uniform PPT adversary $\mathcal{A}$, there exists a negligible function ν such that for every $\lambda \in \mathbb{N}$, the probability that $b' = b$ in the following game is at most $1/2 + \nu(\lambda)$:

1. $(\mathsf{state}, m_0, m_1) \leftarrow \mathcal{A}(1^\lambda)$
2. If $|m_0| \neq |m_1|$ or if $m_0, m_1 \notin \mathcal{M}$ then output $\perp$. Choose $b \overset{\$}{\leftarrow} \{0,1\}$ and sample $c \leftarrow \mathsf{Com}(m_b)$.
3. $b' \leftarrow \mathcal{A}(\mathsf{state}, c)$

4 Random Self-Reducible Languages

In this section we define random self-reducible (RSR) languages and witness-preserving random self-reducible languages, and provide examples of such languages.

4.1 Definitions

Definition 11 (Hard on Average)**.** *A language* L *is said to be* **hard on average** *with respect to a distribution* $\mathcal{D} = \{\mathcal{D}_\lambda\}_{\lambda\in\mathbb{N}}$*, where* $\mathcal{D}_\lambda$ *is over* $\{0,1\}^\lambda \cap L$*, if there exists a distribution* $\bar{\mathcal{D}} = \{\bar{\mathcal{D}}_\lambda\}_{\lambda\in\mathbb{N}}$*, where* $\bar{\mathcal{D}}_\lambda$ *is over* $\{0,1\}^\lambda \setminus L$*, such that* $\mathcal{D} \approx_c \bar{\mathcal{D}}$*.*

Definition 12 (Hard-to-Extract on Average)**.** *A language* $L \in \mathsf{NP}$*, with a corresponding* NP *relation* R_L*, is said to be* **hard-to-extract** *with respect to a distribution* $\mathcal{D} = \{\mathcal{D}_\lambda\}_{\lambda\in\mathbb{N}}$ *if for any poly-size* $\mathcal{A}$ *there exists a negligible function* ν *such that for all* $\lambda \in \mathbb{N}$*,*

$$\Pr_{x \leftarrow \mathcal{D}_\lambda}[w = \mathcal{A}(x) \ : \ (x, w) \in R_L] \leq \nu(\lambda)$$

In what follows we define the notion of a random self-reducible language.

Definition 13 (Random Self Reducibility)**.** *An* NP *language* L *with a corresponding* NP *relation* R_L *is said to be* **random self-reducible** *(RSR) with respect to distribution* $\mathcal{D} = \{\mathcal{D}_\lambda\}_{\lambda\in\mathbb{N}}$*, where* $\mathcal{D}_\lambda$ *is over* $\{0,1\}^\lambda \cap L$*, if there exists a polynomial* p *and a poly-time computable function family* $f = \{f_\lambda\}_{\lambda\in\mathbb{N}}$ *such that for every* $\lambda \in \mathbb{N}$*,*

$$f_\lambda : \{0,1\}^\lambda \times \{0,1\}^{p(\lambda)} \to \{0,1\}^\lambda,$$

and the following two conditions hold.

- *For every $x \in \{0,1\}^\lambda \cap L$ and for $r \xleftarrow{\$} \{0,1\}^{p(\lambda)}$, $f_\lambda(x,r) \approx_s y$, where $y \leftarrow \mathcal{D}_\lambda$.*
- *For every $x \in \{0,1\}^\lambda \setminus L$ and for every $r \in \{0,1\}^{p(\lambda)}$, $f_\lambda(x,r) \notin L$.*

Definition 14 (Witness Preserving Random Self Reducibility)**.** *An* NP *language L with a corresponding* NP *relation R_L is said to be* **witness-preserving random self-reducible** *with respect to distribution $\mathcal{D} = \{\mathcal{D}_\lambda\}_{\lambda\in\mathbb{N}}$, where $\mathcal{D}_\lambda$ is over $\{0,1\}^\lambda \cap L$, if there exists a polynomial p and a poly-time computable function family $f = \{f_\lambda\}_{\lambda\in\mathbb{N}}$ such that for every $\lambda \in \mathbb{N}$,*

$$f_\lambda : \{0,1\}^\lambda \times \{0,1\}^{p(\lambda)} \to \{0,1\}^\lambda$$

and the following two conditions hold.

- *For any $x \in \{0,1\}^\lambda \cap L$ and for $r \xleftarrow{\$} \{0,1\}^{p(\lambda)}$, $f_\lambda(x,r) \approx_s y$, where $y \leftarrow \mathcal{D}_\lambda$.*
- *Let $q(\lambda)$ be the length of w for $(x,w) \in R_{L_\lambda}$. There exist poly-time computable function families $g = \{g_\lambda\}_{\lambda\in\mathbb{N}}$ and $h = \{h_\lambda\}_{\lambda\in\mathbb{N}}$, where for every $\lambda \in \mathbb{N}$*

  $g_\lambda : \{0,1\}^\lambda \times \{0,1\}^{p(\lambda)} \times \{0,1\}^{q(\lambda)} \to \{0,1\}^{q(\lambda)}$ *and* $h_\lambda : \{0,1\}^\lambda \times \{0,1\}^{p(\lambda)} \times \{0,1\}^{q(\lambda)} \to \{0,1\}^{q(\lambda)}$,

  such that the following holds.

  - *For any $x \in \{0,1\}^\lambda \cap L$, any $r \in \{0,1\}^{p(\lambda)}$ and any $w' \in \{0,1\}^{q(\lambda)}$, if $(f_\lambda(x,r), w') \in R_L$ then $(x, g_\lambda(x,r,w')) \in R_L$.*
  - *For any $x \in \{0,1\}^\lambda \cap L$ and any $w \in \{0,1\}^{q(\lambda)}$ such that $(x,w) \in R_L$, it holds that for every $r \in \{0,1\}^{p(\lambda)}$, $(f_\lambda(x,r), h_\lambda(x,r,w)) \in R_L$.*

Note: We will refer to f, g, h as the *reduction functions.*

Remark 4. *We note that the notions of witness-preserving RSR and RSR are incomparable.*

Definition 15 (Instance-Witness Distribution)**.** *Let $L \in$* NP*, let R_L be the corresponding* NP *relation, and let $\mathcal{D} = \{\mathcal{D}_\lambda\}_{\lambda\in\mathbb{N}}$ be a distribution, where $\mathcal{D}_\lambda$ is over L_λ. A distribution $\mathcal{E} = \{\mathcal{E}_\lambda\}_{\lambda\in\mathbb{N}}$ is said to be an instance-witness distribution corresponding to $\mathcal{D}$ if for every (x,w) in the support of $\mathcal{E}_\lambda$ it holds that $(x,w) \in R_{L_\lambda}$, and*

$$\{(x,w) \leftarrow \mathcal{E}_\lambda \ : \ x\}_{\lambda\in\mathbb{N}} \equiv \{y \leftarrow \mathcal{D}_\lambda \ : \ y\}_{\lambda\in\mathbb{N}} \tag{4}$$

4.2 Examples of Random Self-Reducible Languages

4.2.1 Decisional Diffie Hellman

Let $L_{\mathsf{DDH}} = \{L_{\mathsf{DDH},\lambda}\}_{\lambda\in\mathbb{N}}$ be the following language, where for each $\lambda \in \mathbb{N}$ the language $L_{\mathsf{DDH},\lambda}$ is parameterized by a group $\mathbb{G}$ of prime order $p \in [2^{\lambda-1}, 2^\lambda]$ and a generator $g \in \mathbb{G}$:

$$L_{\mathsf{DDH},\lambda} = \{(X,Y,Z) \mid \exists\, x,y \in \mathbb{Z}_p^* \text{ such that } X = g^x \ \wedge\ Y = g^y \ \wedge\ Z = g^{xy}\}$$

Theorem 5. *L_{DDH} is a random self-reducible language (Definition 13) and a witness-preserving random self-reducible language (Definition 14), with respect to the distribution $\mathcal{U}_{\mathsf{DDH}} = \{\mathcal{U}_{\mathsf{DDH},\lambda}\}_{\lambda\in\mathbb{N}}$, where for each $\lambda \in \mathbb{N}$, the distribution $\mathcal{U}_{\mathsf{DDH},\lambda}$ generates (g^x, g^y, g^{xy}) for $x, y \xleftarrow{\$} \mathbb{Z}_p^*$.*

Proof. Consider the poly-time computable function $f_\lambda : L_{\mathsf{DDH},\lambda} \times (\mathbb{Z}_p^*)^2 \to L_{\mathsf{DDH},\lambda}$ defined by

$$f_\lambda((X,Y,Z),(r,s)) = (X^r, Y^s, Z^{rs}).$$

For any $(g^x, g^y, g^{xy}) \in L_{\mathsf{DDH},\lambda}$ and for $r, s \xleftarrow{\$} \mathbb{Z}_p^*$, the tuple $(g^{xr}, g^{ys}, g^{xyrs})$ is distributed according to $\mathcal{U}_{\mathsf{DDH},\lambda}$. Moreover, for every (g^x, g^y, g^z) where $z \neq xy$ it holds that for every $(r,s) \in \mathbb{Z}_p^*$,

$$f((g^x, g^y, g^z),(r,s)) = (g^{xr}, g^{ys}, g^{zrs}) \notin L_{\mathsf{DDH},\lambda}$$

since $zrs \neq xyrs$. This proves that L_{DDH} is random self-reducible.

Also consider poly-time computable functions $g_\lambda, h_\lambda : L_{\mathsf{DDH},\lambda} \times \mathbb{Z}_p^* \times \mathbb{Z}_p^* \to \mathbb{Z}_p^*$, defined as follows:

$$g_\lambda((X,Y,Z),(r,s),(a,b)) = (ar^{-1}, bs^{-1}), \quad h_\lambda((X,Y,Z),(r,s),(x,y)) = (xr, ys)$$

For every $(A,B,C) = f_\lambda((X,Y,Z),(r,s)) = (X^r, Y^s, Z^{rs})$, if $((A,B,C),(a,b)) \in R_{L_{\mathsf{DDH}}}$ then $((X,Y,Z),(ar^{-1}, bs^{-1})) \in R_{L_{\mathsf{DDH}}}$. Similarly, for every $((X,Y,Z),(x,y)) \in R_{L_{\mathsf{DDH}}}$ and every $(r,s) \in \mathbb{Z}_p^*$, it holds that $(f_\lambda((X,Y,Z),(r,s)),(xr,ys))) \in R_{L_{\mathsf{DDH}}}$, giving us witness-preserving random self-reducibility, as desired. □

4.2.2 Discrete Log

Let $L_{\mathsf{DL}} = \{L_{\mathsf{DL},\lambda}\}_{\lambda\in\mathbb{N}}$ be the following language, where for each $\lambda \in \mathbb{N}$ the language $L_{\mathsf{DL},\lambda}$ is parameterized by a group $\mathbb{G}$ of prime order $p \in [2^{\lambda-1}, 2^\lambda]$ and a generator $g \in \mathbb{G}$, and is defined by

$$L_{\mathsf{DL},\lambda} = \{X \mid \exists\ x \in \mathbb{Z}_p^* \text{ such that } X = g^x\}$$

Theorem 6. *L_{DL} is a witness-preserving random self-reducible language (as per Definition 14) with respect to the uniform distribution $\mathcal{U}_{\mathsf{DL}} = \{\mathcal{U}_{\mathsf{DL},\lambda}\}_{\lambda\in\mathbb{N}}$, where for each $\lambda \in \mathbb{N}$ the distribution $\mathcal{U}_{\mathsf{DL},\lambda}$ generates g^x for $x \xleftarrow{\$} \mathbb{Z}_p^*$.*

Proof. Consider the poly-time computable functions $f_\lambda : L_{\mathsf{DL},\lambda} \times \mathbb{Z}_p^* \to L_{\mathsf{DL},\lambda}$ and $g_\lambda, h_\lambda : L_{\mathsf{DL},\lambda} \times \mathbb{Z}_p^* \times \mathbb{Z}_p^* \to \mathbb{Z}_p^*$, defined as follows:

$$f_\lambda(X,r) = X \cdot g^r$$

and

$$g_\lambda(X,r,y) = y - r, \quad h_\lambda(X,s,z) = z + s$$

Note that for every λ and every $X \in L_{\mathsf{DL},\lambda}$, and for $r \xleftarrow{\$} \mathbb{Z}_p^*$, it holds that $f_\lambda(X,r) = X \cdot g^r$ is distributed according to $\mathcal{U}_{\mathsf{DL},\lambda}$. Moreover, for every $Y = f_\lambda(X,r) = X \cdot g^r$ if $(Y,y) \in R_{L_{\mathsf{DL}}}$ then $(X, y-r) \in R_{L_{\mathsf{DL}}}$. Similarly, for every $(X,z) \in R_{L_{\mathsf{DL}}}$ and every $s \in \mathbb{Z}_p$, for $Y = f(X,s) = X \cdot g^s$ it holds that $(Y, z+s) \in R_{L_{\mathsf{DL}}}$, as desired. □

Remark 5. *L_{DL} has an instance-witness distribution $\mathcal{E}_{\mathsf{DL}} = \{\mathcal{E}_{\mathsf{DL},\lambda}\}_{\lambda\in\mathbb{N}}$, defined as follows: For each $\lambda \in \mathbb{N}$, the distribution $\mathcal{E}_{\mathsf{DL},\lambda}$ outputs (g^w, w) for $w \xleftarrow{\$} \mathbb{Z}_p^*$. Note that g^w is distributed according to $\mathcal{U}_{\mathsf{DL},\lambda}$.*

5 Proofs of Ignorance

In this section, we define proofs of ignorance and construct proof of ignorance protocols for random self-reducible languages.

5.1 Definition

We now define the notion of proof-of-ignorance (PoI) and trapdoor PoI for NP languages. Intuitively we want a proof of ignorance π for $x \in L$ to convince the verifier that the prover does not know a witness corresponding to x.

Definition 16 (Proof of Ignorance). *Let $L \in \mathsf{NP}$, let R_L be the corresponding NP-relation, and let $\mathcal{D}$ be a distribution on the instances of L. A* **proof-of-ignorance** *(PoI) proof system for $(L, \mathcal{D})$ consists of a triplet of PPT algorithms $(\mathsf{Setup}, \mathsf{Gen}, \mathsf{Verify})$ with the following syntax:*

Setup $\mathsf{CRS} \leftarrow \mathsf{Setup}(1^\lambda)$*: The setup algorithm takes as input the security parameter and outputs a common reference string CRS.*

Instance Generation $(x, \pi) \leftarrow \mathsf{Gen}(\mathsf{CRS})$*: The generation algorithm takes as input the CRS and outputs an instance $x \in L$ and a proof-of-ignorance π.*

Verification $0/1 \leftarrow \mathsf{Verify}(\mathsf{CRS}, x, \pi)$*: The verification algorithm takes as input the CRS, instance x, and proof π, and outputs 0 or 1.*

We require the following properties to hold.

Completeness. *We require:*

- $\{\mathsf{CRS} \leftarrow \mathsf{Setup}(1^\lambda) \ ; \ (x, \pi) \leftarrow \mathsf{Gen}(\mathsf{CRS}) \ : \ x\}_{\lambda \in \mathbb{N}} \approx_s \{y \leftarrow \mathcal{D}_\lambda \ : \ y\}_{\lambda \in \mathbb{N}}$.
- *For all $\lambda \in \mathbb{N}$,*

$$\Pr[\mathsf{CRS} \leftarrow \mathsf{Setup}(1^\lambda) \ ; \ (x, \pi) \leftarrow \mathsf{Gen}(\mathsf{CRS}) \ : \ \mathsf{Verify}(\mathsf{CRS}, x, \pi) = 1] = 1.$$

Soundness. *There exists a PPT algorithm Setup' such that*

- $\{\mathsf{CRS} \leftarrow \mathsf{Setup}(1^\lambda) \ : \ \mathsf{CRS}\}_{\lambda \in \mathbb{N}}, \approx_c \{\mathsf{CRS}' \leftarrow \mathsf{Setup}'(1^\lambda) \ : \ \mathsf{CRS}'\}_{\lambda \in \mathbb{N}}$
- *For any all-powerful $\mathcal{A}^*$, there exists a negligible function ν such that for all $\lambda \in \mathbb{N}$,*

$$\Pr[\mathsf{CRS}' \leftarrow \mathsf{Setup}'(1^\lambda) \ ; \ (x, \pi) \leftarrow \mathcal{A}^*(\mathsf{CRS}') \ : \ \mathsf{Verify}(\mathsf{CRS}', x, \pi) = 1 \ \wedge \ x \in L] \leq \nu(\lambda)$$

Definition 17 (Trapdoor Proof of Ignorance). *Let $L \in \mathsf{NP}$, let R_L be the corresponding NP-relation, and let $\mathcal{D}$ be a distribution over instances in L. A* **trapdoor proof-of-ignorance** *(td-PoI) proof system for $(L, \mathcal{D})$ consists of a tuple of PPT algorithms $(\mathsf{Setup}, \mathsf{Gen}, \mathsf{Verify}, \mathsf{Witness})$ with the following syntax (we note that Gen and Verify are identical to those defined in Definition 16):*

Setup $(\mathsf{CRS}, \mathsf{td}) \leftarrow \mathsf{Setup}(1^\lambda)$*: The setup algorithm takes as input the security parameter and outputs a common reference string CRS together with a corresponding trapdoor td.*

Instance Generation $(x, \pi) \leftarrow \mathsf{Gen}(\mathsf{CRS})$*: The generation algorithm takes as input the CRS and outputs an instance $x \in L$ and a proof of ignorance π.*

Verification $0/1 \leftarrow \mathsf{Verify}(\mathsf{CRS}, x, \pi)$*: The verification algorithm takes as input the CRS, instance x, and proof π, and outputs 0 or 1.*

Witness Generation $w \leftarrow \mathsf{Witness}(\mathsf{CRS}, \mathsf{td}, x, \pi)$*: The witness generation algorithm takes as input the CRS together with a corresponding trapdoor, along with an instance x and a proof π, and outputs a string w.*

We require the following properties to hold.

Completeness *We require:*

- $\{(\mathsf{CRS}, \mathsf{td}) \leftarrow \mathsf{Setup}(1^\lambda)\ ;\ (x, \pi) \leftarrow \mathsf{Gen}(\mathsf{CRS})\ :\ x\}_{\lambda\in\mathbb{N}} \approx_s \{y \leftarrow \mathcal{D}_\lambda\ :\ y\}_{\lambda\in\mathbb{N}}$.
- *For all* $\lambda \in \mathbb{N}$,
$$\Pr[\mathsf{CRS} \leftarrow \mathsf{Setup}(1^\lambda)\ ;\ (x, \pi) \leftarrow \mathsf{Gen}(\mathsf{CRS})\ :\ \mathsf{Verify}(\mathsf{CRS}, x, \pi) = 1] = 1.$$
- *For every* $(\mathsf{CRS}, \mathsf{td})$ *in the image of* $\mathsf{Setup}(1^\lambda)$ *and for any* (x, π)*, if* $\mathsf{Verify}(\mathsf{CRS}, x, \pi) = 1$ *then*
$$\mathsf{Witness}(\mathsf{CRS}, \mathsf{td}, x, \pi) \in R_L(x).$$

Soundness *For any poly-size* $\mathcal{A}^*$*, there exists a negligible function* ν *such that for all* $\lambda \in \mathbb{N}$,

$$\Pr[\mathsf{CRS} \leftarrow \mathsf{Setup}(1^\lambda)\ ;\ (x, w, \pi) \leftarrow \mathcal{A}^*(\mathsf{CRS})\ :\ \mathsf{Verify}(\mathsf{CRS}, x, \pi) = 1\ \wedge\ (x, w) \in R_L] \leq \nu(\lambda)$$

Remark 6. *We note that the notions of PoI and trapdoor PoI are incomparable.*

Definition 18 (T-security)**.** *A trapdoor proof of ignorance (*td*-PoI) proof system for* $(L, \mathcal{D})$ *(as in Definition 17) is said to be* T*-secure if soundness holds for all adversaries of size* $\mathsf{poly}(T)$*.*

5.2 Constructions

We now construct a PoI protocol for random self-reducible languages and trapdoor PoI protocol for witness-preserving random self-reducible languages.

Proof of Ignorance for RSR Languages: Let L be hard on average and random self-reducible with respect to distribution $\mathcal{D} = \{\mathcal{D}_\lambda\}_{\lambda\in\mathbb{N}}$ (as per Definitions 11 and 13, respectively). Let

$$f_\lambda : \{0,1\}^\lambda \times \{0,1\}^{p(\lambda)} \to \{0,1\}^\lambda$$

be the corresponding reduction function. The PoI protocol for L is described as follows:

$\mathsf{Setup}(1^\lambda)$: Choose $z \leftarrow \mathcal{D}_\lambda$. Output $\mathsf{CRS} = (z, f_\lambda)$.

$\mathsf{Gen}(\mathsf{CRS})$: Choose $r \overset{\$}{\leftarrow} \{0,1\}^{p(\lambda)}$ and compute $x = f_\lambda(z, r)$. Output (x, π) where $\pi = r$.

$\mathsf{Verify}(\mathsf{CRS}, x, \pi)$: Output 1 if and only if $x = f_\lambda(z, \pi)$ where z is part of the CRS.

Theorem 7. *Let* L *be hard on average and random self-reducible with respect to distribution* $\mathcal{D} = \{\mathcal{D}_\lambda\}_{\lambda\in\mathbb{N}}$ *(as per Definitions 11 and 13, respectively). The protocol described above is a Proof-of-Ignorance proof system for* $(L, \mathcal{D})$ *as per Definition 16.*

Proof. Completeness is straightforward: $\mathsf{Gen}(\mathsf{CRS})$ outputs (x, π) where $x = f_\lambda(z, r)$ and $\pi = r$, and $\mathsf{Verify}(\mathsf{CRS}, x, \pi)$ checks that indeed $x = f_\lambda(z, \pi)$. Also from the definition of random self-reducibility of L, for every $x \in \{0,1\}^\lambda \cap L$ and for $r \overset{\$}{\leftarrow} \{0,1\}^{p(\lambda)}$, $f_\lambda(x, r) \approx_s y$ where $y \leftarrow \mathcal{D}_\lambda$, as desired.

Now we prove soundness. By definition, the fact that L is hard on average with respect to distribution $\mathcal{D} = \{\mathcal{D}_\lambda\}_{\lambda\in\mathbb{N}}$, implies that there exists a distribution $\bar{\mathcal{D}} = \{\bar{\mathcal{D}}_\lambda\}_{\lambda\in\mathbb{N}}$, where $\bar{\mathcal{D}}_\lambda$ is over $\{0,1\}^\lambda \setminus L$, such that $\mathcal{D} \approx_c \bar{\mathcal{D}}$. Define Setup' as follows: On input the security parameter λ, output $\mathsf{CRS}' = (\bar{z}, f_\lambda)$ where $\bar{z} \leftarrow \bar{\mathcal{D}}_\lambda$ and f_λ is the reduction function of L as before. Suppose for contradiction, there exists $\mathcal{A}^*$ and polynomial s such that for infinitely many $\lambda \in \mathbb{N}$,

$$\Pr[(x, \pi) = \mathcal{A}^*(\mathsf{CRS}')\ :\ \mathsf{Verify}(\mathsf{CRS}', x, \pi) = 1\ \wedge\ x \in L] > \frac{1}{s(\lambda)}$$

where the probability is over $\mathsf{CRS}' \leftarrow \mathsf{Setup}'(1^\lambda)$. This implies that for infinitely many $\lambda \in \mathbb{N}$,

$$\Pr[(x,\pi) = \mathcal{A}^*(\bar{z}, f_\lambda) \ : \ x = f_\lambda(\bar{z}, r) \ \wedge \ x \in L] > \frac{1}{s(\lambda)}$$

where the probability is over $\bar{z} \leftarrow \bar{\mathcal{D}}_\lambda$, contradicting the fact that for every $\bar{z} \in \{0,1\}^\lambda \setminus L$ and every $r \in \{0,1\}^{p(\lambda)}$, $f_\lambda(\bar{z}, r) \notin L$.

□

Trapdoor Proof of Ignorance for Witness-preserving RSR Languages: Let L be hard-to-extract and witness-preserving random self-reducible with respect to distribution $\mathcal{D} = \{\mathcal{D}_\lambda\}_{\lambda \in \mathbb{N}}$ (as per Definitions 12 and 14, respectively). Let $\mathcal{E} = \{\mathcal{E}_\lambda\}_{\lambda \in \mathbb{N}}$ be a corresponding instance-witness distribution on R_L as per Definition 15. Let

$$f_\lambda : \{0,1\}^\lambda \times \{0,1\}^{p(\lambda)} \to \{0,1\}^\lambda \text{ and } g_\lambda, h_\lambda : \{0,1\}^\lambda \times \{0,1\}^{p(\lambda)} \times \{0,1\}^{q(\lambda)} \to \{0,1\}^{q(\lambda)}$$

be the corresponding reduction functions. The trapdoor PoI protocol for L is described as follows:

$\mathsf{Setup}(1^\lambda)$: Choose $(z, w) \leftarrow \mathcal{E}_\lambda$. Output $\mathsf{CRS} = (z, f_\lambda, h_\lambda)$ and $\mathsf{td} = w$.

$\mathsf{Gen}(\mathsf{CRS})$: Choose $r \overset{\$}{\leftarrow} \{0,1\}^{p(\lambda)}$ and compute $x = f_\lambda(z, r)$. Output (x, π) where $\pi = r$.

$\mathsf{Verify}(\mathsf{CRS}, x, \pi)$: Output 1 if and only if $x = f_\lambda(z, \pi)$ where z is part of the CRS.

$\mathsf{Witness}(\mathsf{CRS}, \mathsf{td}, x, \pi)$: Output $h_\lambda(x, \pi, \mathsf{td})$.

Theorem 8. *Let L be hard-to-extract and witness-preserving random self-reducible with respect to distribution $\mathcal{D} = \{\mathcal{D}_\lambda\}_{\lambda \in \mathbb{N}}$ (as per Definition 12 and 14, respectively). Assume that there exists an instance witness distribution $\mathcal{E} = \{\mathcal{E}_\lambda\}_{\lambda \in \mathbb{N}}$ corresponding to $\mathcal{D}$ that is efficiently sampleable. Then the protocol described above is a Trapdoor Proof of Ignorance proof system for $(L, \mathcal{D})$ as per Definition 17.*

Proof. Completeness is straightforward, we thus focus on proving soundness. To this end, suppose for contradiction that there exists a poly-size $\mathcal{A}^*$ and a polynomial s such that for infinitely many $\lambda \in \mathbb{N}$,

$$\Pr[(x, w, \pi) = \mathcal{A}^*(\mathsf{CRS}) \ : \ \mathsf{Verify}(\mathsf{CRS}, x, \pi) = 1 \ \wedge \ (x, w) \in R_L] > \frac{1}{s(\lambda)} \tag{5}$$

where the probability is over $\mathsf{CRS} \leftarrow \mathsf{Setup}(1^\lambda)$. The fact that L is *hard-to-extract* with respect to the distribution $\mathcal{D}$ implies that for every poly-size B^* there exists a negligible function μ such that for all $\lambda \in \mathbb{N}$,

$$\Pr_{z \leftarrow D_\lambda}[w = B^*(z) \ : \ (z, w) \in R_L] \leq \mu(\lambda) \tag{6}$$

We use A^* from Equation (5) to construct a poly-size B^* that contradicts Equation (6). B^* on input z, sets $\mathsf{CRS} = (z, f_\lambda, h_\lambda)$, computes $(x, \pi, w) = A^*(\mathsf{CRS})$, and outputs $g_\lambda(x, \pi, w)$. By Equation (5), for infinitely many $\lambda \in \mathbb{N}$,

$$\Pr[(x, w, \pi) = \mathcal{A}^*(\mathsf{CRS}) \ : \ x = f_\lambda(z, \pi) \ \wedge \ (x, w) \in R_L] > \frac{1}{s(\lambda)}$$

where $\mathsf{CRS} = (z, f_\lambda, h_\lambda)$ and the probability is over $z \leftarrow \mathcal{D}_\lambda$. By the definition of witness preserving random self-reducibility, if $(f_\lambda(z, \pi), w) \in R_L$ then $(z, g_\lambda(x, \pi, w)) \in R_L$. Hence, we conclude that for infinitely many $\lambda \in \mathbb{N}$,

$$\Pr_{z \overset{\$}{\leftarrow} D_\lambda}[w = B^*(z) \ : \ (z, w) \in R_L] > \frac{1}{s(\lambda)},$$

contradicting Equation (6).

□

6 Witness Hiding Arguments from Proofs of Ignorance

In this section, we show how to use a PoI proof system to construct a 2-message witness hiding argument for NP with adaptive soundness.

6.1 Ingredients

We first describe the ingredients we use in our witness hiding protocol. We assume that there exists a super-polynomial function $T = T(\lambda)$, for which the following primitives exists:

- A T-secure Trapdoor Proof of Ignorance (td-PoI) system for any $(L', \mathcal{D}')$, as defined in Definition 17 (Section 5.1), denoted by

$$(\mathsf{PoI.Setup}, \mathsf{PoI.Gen}, \mathsf{PoI.Verify}, \mathsf{PoI.Witness}).$$

- A T-secure non-interactive witness indistinguishable (NIWI) proof system with perfect soundness,[10] as defined in Definition 6 (Section 3.1), denoted by

$$(\mathsf{NIWI.Prove}, \mathsf{NIWI.Verify})$$

- A T-secure bit-wise rerandomizable encryption scheme that is strong KDM secure, as defined in Definitions 7, 8, and 9 (Section 3.2), denoted by

$$(\mathsf{PKE.Gen}, \mathsf{PKE.Enc}, \mathsf{PKE.Dec}).$$

- A non-interactive statistically binding commitment scheme, as defined in Definition 10 (Section 3.2), denoted by Com.

  We assume that the hiding property of Com can be broken in time $T = T(\lambda)$. Namely, we assume that in time $\mathsf{poly}(T)$ one can brute-force break the commitment scheme; i.e., there exist a T-time adversary $\mathcal{A}$ such that for every $m \in \mathcal{M}$ and every $r \xleftarrow{\$} \{0,1\}^\lambda$,

$$\mathcal{A}(\mathsf{Com}(m,r)) = (m, r') \ \text{ s.t. } \ \mathsf{Com}(m,r') = \mathsf{Com}(m,r).$$

Theorem 9. *Assuming the ingredients above there exists a two-message WH argument for* NP *with adaptive soundness.*

Corollary 2. *Let $T = n^{\omega(1)}$. There exists a two-message WH protocol for* NP *with adaptive soundness, assuming the existence of a T-secure rerandomizable encryption that is strong KDM secure, and assuming the T-security of DLIN.*

6.2 The Protocol Description

We next describe our 2-message witness hiding argument for any $L \in \mathsf{NP}$ and any distribution $\mathcal{D}$ over pairs in R_L.

[10] The requirement of perfect soundness is not needed, and is only made for simplicity. We note that the NIWI proof system based on DLIN [GOS06] indeed has perfect soundness.

- **The verifier's message:** On input 1^λ, the verifier V_1 does the following:

  1. For every $b \in \{0,1\}$, do the following: Sample $(\mathsf{pk}_b, \mathsf{sk}_b) \leftarrow \mathsf{PKE.Gen}(1^\lambda)$, choose at random $r_b^1, r_b^2 \leftarrow \{0,1\}^\lambda$, and compute $(\mathsf{CRS}_b, \mathsf{td}_b) = \mathsf{Pol.Setup}(1^\lambda; r_b^1)$ and $\mathsf{ct}_b = \mathsf{PKE.Enc}_{\mathsf{pk}_b}(\mathsf{td}_b; r_b^2)$.
  2. Consider the NP language

     $$L^* \triangleq \{(\mathsf{CRS}, \mathsf{pk}, \mathsf{ct}) : \exists(\mathsf{td}, r^1, r^2) \text{ s.t.}(\mathsf{CRS}, \mathsf{td}) = \mathsf{Pol.Setup}(1^\lambda, r^1) \wedge \mathsf{ct} = \mathsf{PKE.Enc}_{\mathsf{pk}}(\mathsf{td}, r^2)\}$$

     and consider the NP language

     $$L^*_{\mathsf{OR}} \triangleq \{(x_1^*, x_2^*) : \exists w^* \text{ s.t. } (x_1^*, w^*) \in L^* \ \vee (x_1^*, w^*) \in L^*\}$$

  3. For every $b \in \{0,1\}$, let $x_b^* = (\mathsf{CRS}_b, \mathsf{pk}_b, \mathsf{ct}_b)$. Choose $b^* \xleftarrow{\$} \{0,1\}$ and let $w^* = (\mathsf{td}_{b^*}, r_{b^*}^1, r_{b^*}^2)$. Generate a NIWI proof $\pi_{\mathsf{NIWI}} \leftarrow \mathsf{NIWI.Prove}((x_0^*, x_1^*), w^*)$.

  Output $(\mathsf{pp}, \mathsf{st})$ where

  $$\mathsf{pp} = (x_0^*, x_1^*, \pi_{\mathsf{NIWI}}) = \big(\{(\mathsf{CRS}_b, \mathsf{pk}_b, \mathsf{ct}_b)\}_{b\in\{0,1\}}, \pi_{\mathsf{NIWI}}\big)$$

  and $\mathsf{st} = (\mathsf{sk}_0, \mathsf{sk}_1)$.

- **The prover's message:**

  On input $(1^\lambda, x, w)$, and public parameters

  $$\mathsf{pp} = (x_0^*, x_1^*, \pi_{\mathsf{NIWI}}) = \big(\{(\mathsf{CRS}_b, \mathsf{pk}_b, \mathsf{ct}_b)\}_{b\in\{0,1\}}, \pi_{\mathsf{NIWI}}\big),$$

  the prover does the following:

  1. Check that $\mathsf{NIWI.Verify}((x_0^*, x_1^*), \pi_{\mathsf{NIWI}}) = 1$. If this condition is not satisfied then abort.
  2. For every $b \in \{0,1\}$, compute $(x_b', \pi_b') \leftarrow \mathsf{Pol.Gen}(\mathsf{CRS}_b)$, choose at random $r_b', s_b' \leftarrow \{0,1\}^\lambda$ and compute $c_b' = \mathsf{Com}(\pi_b'; r_b')$ and $\mathsf{ct}_b' = \mathsf{PKE.Enc}_{\mathsf{pk}_b}((\pi_b', r_b'); s_b')$.
  3. Compute $c' \leftarrow \mathsf{Com}(0)$.
  4. Consider the language

     $$L_{\mathsf{Pol}} = \big\{(x, x_0', x_1', c') \mid \exists(w, b, w', r) \text{ s.t. } \big((x, w) \in R_L\big)\vee \\ \big(\big(c' = \mathsf{Com}(w'; r)\big) \wedge \big((x_b', w') \in R_{L'}\big)\big)\big\}.$$

     Generate a NIWI proof π'_{NIWI} for $(x, x_0', x_1', c') \in L_{\mathsf{Pol}}$, using a witness w for x.
  5. Output $\big(\{(x_b', c_b', \mathsf{ct}_b')\}_{b\in\{0,1\}}, c', \pi'_{\mathsf{NIWI}}\big)$.

- **The verifier's verdict:** V_2 on input $\big(1^\lambda, \mathsf{pp}, \mathsf{st}, (x, \mathsf{msg})\big)$ outputs 1, where

  $$\mathsf{msg} = \big(\{(x_b', c_b', \mathsf{ct}_b')\}_{b\in\{0,1\}}, c', \pi'_{\mathsf{NIWI}}\big)$$

  if and only if the all the following checks pass.

  1. For every $b \in \{0,1\}$, compute $(\pi_b', r_b') = \mathsf{PKE.Dec}_{\mathsf{sk}_b}(\mathsf{ct}_b')$, and check that $c_b' = \mathsf{Com}(\pi_b'; r_b')$ and $\mathsf{Pol.Verify}(x_b', \pi_b', \mathsf{CRS}_b) = 1$.
  2. Check that $\mathsf{NIWI.Verify}\big((x, x_0', x_1', c'), \pi'_{\mathsf{NIWI}}\big) = 1$.

6.3 The Analysis

In this section we prove that the protocol defined in Section 6.2 satisfies Theorem 9.

Proof of Theorem 9. In what follows, we prove that the protocol defined in Section 6.2 satisfies the completeness, soundness and witness hiding properties.

Completeness. Completeness follows directly from the completeness of the underlying primitives.

Adaptive Soundness. Assume for contradiction that there exists a non-uniform poly-size cheating prover P^*, a polynomial s, and an infinite set $\Lambda \subseteq \mathbb{N}$, such that for every $\lambda \in \Lambda$,

$$\Pr\Big[\big((P^*, V)(1^\lambda) = 1\big) \ \wedge \ \big(x \notin L\big)\Big] > \frac{1}{s(\lambda)}. \tag{7}$$

where the probability is over $\mathsf{pp} \leftarrow V(1^\lambda)$ and where $(x, \mathsf{msg}) = P^*(\mathsf{pp})$. Parse

$$\mathsf{msg} = \Big(\{(x'_b, c'_b, \mathsf{ct}'_b)\}_{b\in\{0,1\}}, c', \pi'_{\mathsf{NIWI}}\Big).$$

We use P^* to construct a $\mathsf{poly}(T)$-size adversary $\mathcal{A}$ that takes as input CRS generated according to $\mathsf{PoI.Setup}(1^\lambda)$, and outputs a tuple (x', w', π') such that $(x', w') \in L'$ and $\mathsf{PoI.Verify}(\mathsf{CRS}, x', \pi') = 1$, contradicting the T-security of the td-PoI system. The algorithm $\mathcal{A}$ on input CRS, does the following:

1. Choose at random $b^* \leftarrow \{0,1\}$, and set $\mathsf{CRS}_{1-b^*} = \mathsf{CRS}$.
2. Choose at random $r^1_{b^*} \leftarrow \{0,1\}^\lambda$ and compute $(\mathsf{CRS}_{b^*}, \mathsf{td}_{b^*}) = \mathsf{PoI.Setup}(1^\lambda; r^1_{b^*})$.
3. Generate $(\mathsf{pk}_0, \mathsf{sk}_0), (\mathsf{pk}_1, \mathsf{sk}_1) \leftarrow \mathsf{PKE.Gen}(1^\lambda)$.
4. Choose at random $r^2_{b^*} \leftarrow \{0,1\}^\lambda$ and compute $\mathsf{ct}'_{b^*} = \mathsf{PKE.Enc}_{\mathsf{pk}_{b^*}}(\mathsf{td}_{b^*}; r^2_{b^*})$.
5. Generate $\mathsf{ct}'_{1-b^*} \leftarrow \mathsf{PKE.Enc}_{\mathsf{pk}_{1-b^*}}(0)$.
6. Let $x^*_0 = (\mathsf{CRS}_0, \mathsf{pk}_0, \mathsf{ct}_0)$ and $x^*_1 = (\mathsf{CRS}_1, \mathsf{pk}_1, \mathsf{ct}_1)$, and let $w^* = (\mathsf{td}_{b^*}, r^1_{b^*}, r^2_{b^*})$
7. Compute $\pi_{\mathsf{NIWI}} \leftarrow \mathsf{NIWI.Prove}(x^*_0, x^*_1, w^*)$.
8. Let $\mathsf{pp} = (x^*_0, x^*_1, \pi_{\mathsf{NIWI}})$.
9. Compute $\big(x, \mathsf{msg}\big) = P^*\left(\mathsf{pp}\right)$, and parse $\mathsf{msg} = \Big(\{(x'_b, c'_b, \mathsf{ct}'_b)\}_{b\in\{0,1\}}, c', \pi'_{\mathsf{NIWI}}\Big)$.
10. Run in time $\mathsf{poly}(T)$ to find (w', r') such that $c' = \mathsf{Com}(w'; r')$, and to find $(\pi'_{1-b^*}, r'_{1-b^*})$ such that $c'_{1-b^*} = \mathsf{Com}(\pi'_{1-b^*}; r'_{1-b^*})$.[11]
11. Output $(x'_{1-b^*}, w', \pi'_{1-b^*})$.

We prove that there exists a polynomial q such that for every $\lambda \in \Lambda$,

$$\Pr\big[\mathcal{A}(\mathsf{CRS}) = (x', w', \pi') \text{ s.t. } \big((x', w') \in R_{L'}\big) \wedge \big(\mathsf{PoI.Verify}(\mathsf{CRS}, x', \pi') = 1\big)\big] \geq \frac{1}{q(\lambda)}, \tag{8}$$

contradicting the T-security of the PoI scheme.

[11] We note that the randomness computed in this step may not be the actual randomness used by P^*. This abuse of notation (or notational overload) is only to avoid cluttering on notation, and is of no significance.

To this end, consider the following dishonest verifier $V_{\mathcal{A},b^*}$ that generates his first message with the same distribution as $\mathcal{A}$, while fixing his random bit choice to be b^*. Moreover, it outputs 1 if and only if the NIWI proof given by the prover is accepting, and for $(\pi'_{b^*}, r'_{b^*}) = \mathsf{PKE.Dec}_{\mathsf{sk}_{b^*}}(\mathsf{ct}'_{b^*})$ it holds that

$$c'_{b^*} = \mathsf{Com}(\pi'_{b^*}; r'_{b^*}) \text{ and } \mathsf{Pol.Verify}(x'_{b^*}, \pi'_{b^*}, \mathsf{CRS}_{b^*}) = 1.$$

Namely, $V_{\mathcal{A},b^*}$ does the same checks as the honest verifier, except that he does not check the conditions corresponding to $1 - b^*$.

Claim 1. *For every poly-size adversary $\mathcal{B}$, there exists a negligible function ν such that for every $\lambda \in \Lambda$,*

$$\left| \Pr\Big[\mathcal{B}(\mathsf{pp}, (x, \mathsf{msg}), b, \mathsf{sk}_b, w', r) = 1\Big] - \Pr\Big[\mathcal{B}(\mathsf{pp}_{\mathcal{A},b}, (x, \mathsf{msg}), b, \mathsf{sk}_b, w', r) = 1\Big] \right| \leq \nu(\lambda)$$

where the left probability is over $\mathsf{pp} = (x_0^*, x_1^*, \pi_{\mathsf{NIWI}}) \leftarrow V(1^\lambda)$, where $(x, \mathsf{msg}) = P^*(\mathsf{pp})$, and $b \in \{0,1\}$ is such that π_{NIWI} is generated with a witness corresponding to x_b^*, and where sk_b is the secret key corresponding to pk_b where $x_b^* = (\mathsf{CRS}_b, \mathsf{pk}_b, \mathsf{ct}_b)$. The right probability is over $b \xleftarrow{\$} \{0,1\}$ and $\mathsf{pp}_{\mathcal{A},b} = (x_0^*, x_1^*, \pi_{\mathsf{NIWI}}) \leftarrow V_{\mathcal{A},b}(1^\lambda)$, where $(x, \mathsf{msg}) = P^*(\mathsf{pp}_{\mathcal{A},b})$, and where sk_b is the secret key corresponding to pk_b where $x_b^* = (\mathsf{CRS}_b, \mathsf{pk}_b, \mathsf{ct}_b)$. In both probabilities (w', r) satisfies $c' = \mathsf{Com}(w'; r)$, where c' is part of msg.[12]

Proof. Suppose for contradiction there exists a poly-size adversary $\mathcal{B}$, polynomial p such that for infinitely many $\lambda \in \Lambda$

$$\Pr\Big[\mathcal{B}(\mathsf{pp}, (x, \mathsf{msg}), b, \mathsf{sk}_b, w', r) = 1\Big] - \Pr\Big[\mathcal{B}(\mathsf{pp}_{\mathcal{A},b}, (x, \mathsf{msg}), b, \mathsf{sk}_b, w', r) = 1\Big] > \frac{1}{p(\lambda)} \tag{9}$$

We use $\mathcal{B}$ to construct a $\mathsf{poly}(T)$-size adversary $\mathcal{M}$ that contradicts the T-security of the encryption scheme as per Lemma 1.

Algorithm $\mathcal{M}(1^\lambda)$ does the following:

1. For every $b \in \{0,1\}$, choose $r_b^1 \xleftarrow{\$} \{0,1\}^\lambda$ and compute $(\mathsf{CRS}_b, \mathsf{td}_b) = \mathsf{Pol.Setup}(1^\lambda; r_b^1)$.

2. Choose at random $d \xleftarrow{\$} \{0,1\}$, set $m_0 = 0$ and $m_1 = \mathsf{td}_{1-d}$ such that $|m_0| = |m_1|$, and send m_0 and m_1 as challenge messages.

3. Upon receiving from the challenger a pair $(\mathsf{pk}, \mathsf{ct})$, where $\mathsf{ct} = \mathsf{PKE.Enc}_{\mathsf{pk}}(m_{d^*})$ for a random $d^* \xleftarrow{\$} \{0,1\}$, do the following:

   (a) Generate $(\mathsf{pk}_d, \mathsf{sk}_d) \leftarrow \mathsf{PKE.Gen}(1^\lambda)$, and let $\mathsf{pk}_{1-d} = \mathsf{pk}$.

   (b) Choose $r_d^2 \xleftarrow{\$} \{0,1\}^\lambda$, compute $\mathsf{ct}_d = \mathsf{PKE.Enc}_{\mathsf{pk}_d}(\mathsf{td}_d; r_d^2)$, and let $\mathsf{ct}_{1-d} = \mathsf{ct}$.

   (c) Let $w = (\mathsf{td}_d, r_d^1, r_d^2)$, and compute $\pi_{\mathsf{NIWI}} \leftarrow \mathsf{NIWI.Prove}\big(\{(\mathsf{CRS}_b, \mathsf{pk}_b, \mathsf{ct}_b)\}_{b \in \{0,1\}}, w\big)$.

   (d) Let $\mathsf{pp} = (\{(\mathsf{CRS}_b, \mathsf{pk}_b, \mathsf{ct}_b)\}_{b\in\{0,1\}}, \pi_{\mathsf{NIWI}})$, compute $(x, \mathsf{msg}) = P^*(\mathsf{pp})$, and parse

   $$\mathsf{msg} = \Big(\{(x'_b, c'_b, \mathsf{ct}'_b)\}_{b\in\{0,1\}}, c', \pi'_{\mathsf{NIWI}}\Big).$$

   (e) Run in time $\mathsf{poly}(T)$ to find (w', r) such that $c' = \mathsf{Com}(w'; r)$.

   (f) Output $\mathcal{B}(\mathsf{pp}, (x, \mathsf{msg}), d, \mathsf{sk}_d, w', r)$.

[12] Recall that $\mathsf{msg} = \Big(\{(x'_b, c'_b, \mathsf{ct}'_b)\}_{b\in\{0,1\}}, c', \pi'_{\mathsf{NIWI}}\Big)$.

Note that if $d^* = 1$ then the input to $\mathcal{B}$ is distributed exactly as in the left side of Equation (9), whereas if $d^* = 0$ the input to $\mathcal{B}$ is distributed exactly as in the right side of Equation (9).

$$\begin{aligned}
&\Pr\left[\mathcal{M}(\mathsf{pk}, \mathsf{ct}) = d^*\right] = \\
&\Pr\left[\mathcal{B}(\mathsf{pp}, (x, \mathsf{msg}), d, \mathsf{sk}_d, w', r) = d^*\right] = \\
&\frac{1}{2} \cdot \Pr\left[\mathcal{B}(\mathsf{pp}, (x, \mathsf{msg}), d, \mathsf{sk}_d, w', r) = d^* \;\middle|\; d^* = 1\right] + \frac{1}{2} \cdot \Pr\left[\mathcal{B}(\mathsf{pp}, (x, \mathsf{msg}), d, \mathsf{sk}_d, w', r) = d^* \;\middle|\; d^* = 0\right] = \\
&\frac{1}{2} \cdot \Pr\left[\mathcal{B}(\mathsf{pp}, (x, \mathsf{msg}), d, \mathsf{sk}_d, w', r) = 1\right] + \frac{1}{2} \cdot \Pr\left[\mathcal{B}(\mathsf{pp}_{\mathcal{A},d}, (x, \mathsf{msg}), d, \mathsf{sk}_d, w', r) = 0\right] = \\
&\frac{1}{2} \cdot \Pr\left[\mathcal{B}(\mathsf{pp}, (x, \mathsf{msg}), d, \mathsf{sk}_d, w', r) = 1\right] + \frac{1}{2} \cdot \Big(1 - \Pr\left[\mathcal{B}(\mathsf{pp}_{\mathcal{A},d}, (x, \mathsf{msg}), d, \mathsf{sk}_d, w', r) = 1\right]\Big) = \\
&\frac{1}{2} + \frac{1}{2} \cdot \Big(\Pr\left[\mathcal{B}(\mathsf{pp}, (x, \mathsf{msg}), d, \mathsf{sk}_d, w', r) = 1\right] - \Pr\left[\mathcal{B}(\mathsf{pp}_{\mathcal{A},d}, (x, \mathsf{msg}), d, \mathsf{sk}_d, w', r) = 1\right]\Big) \geq \\
&\frac{1}{2} + \frac{1}{2p(\lambda)}
\end{aligned}$$

where the last inequality follows from Equation (9) (for infinitely many $\lambda \in \Lambda$). This contradicts the T-semantic security of the underlying encryption, as desired. □

Let $\mathsf{pp}_{\mathcal{A},b^*} \leftarrow V_{\mathcal{A},b^*}(1^\lambda)$ and let $(x, \mathsf{msg}) \leftarrow P^*(\mathsf{pp}_{\mathcal{A},b^*})$. Parse $\mathsf{msg} = \big(\{(x'_b, c'_b, \mathsf{ct}'_b)\}_{b\in\{0,1\}}, c', \pi'_{\mathsf{NIWI}}\big)$. For every $b \in \{0,1\}$ let E_b be the event that there exists (w', r) such that

$$c' = \mathsf{Com}(w'; r) \quad \text{and} \quad (x'_b, w') \in R_{L'}.$$

Claim 2. *There exists a polynomial p such that for every $\lambda \in \Lambda$,*

$$\Pr\Big[\big(E_0 \vee E_1\big) \;\wedge\; \big((P^*, V_{\mathcal{A},b^*})(1^\lambda) = 1\big)\Big] \geq \frac{1}{p(\lambda)},$$

where the probability is over $b^* \overset{\$}{\leftarrow} \{0,1\}$ and over the randomness of $V_{\mathcal{A},b^*}$.

Proof. By our contradiction assumption (Equation (7)), and by the soundness of the NIWI proof system, there exists a negligible function μ such that for every $\lambda \in \Lambda$,

$$\Pr\Big[\big(E_0 \vee E_1\big) \;\wedge\; \big((P^*, V)(1^\lambda) = 1\big)\Big] \geq \frac{1}{s(\lambda)} - \mu(\lambda),$$

where the probability is over the randomness of V. This, together with Claim 1, implies that there exists a negligible function ν such that for every $\lambda \in \Lambda$,

$$\Pr\Big[\big(E_0 \vee E_1\big) \;\wedge\; \big((P^*, V_{\mathcal{A},b^*})(1^\lambda) = 1\big)\Big] \geq \frac{1}{s(\lambda)} - \nu(\lambda),$$

where the probability is over $b^* \overset{\$}{\leftarrow} \{0,1\}$ and over the randomness of $V_{\mathcal{A},b^*}$, as desired. □

Claim 3. *There exists a polynomial q such that for every $\lambda \in \Lambda$,*

$$\Pr\Big[E_{1-b} \;\wedge\; \big((P^*, V_{\mathcal{A},b})(1^\lambda) = 1\big)\Big] \geq \frac{1}{q(\lambda)},$$

where the probability is over $b \overset{\$}{\leftarrow} \{0,1\}$ and the randomness of $V_{\mathcal{A},b}$.

Proof. Suppose for contradiction that there exists a negligible function μ and an infinite set $\Lambda_0 \subseteq \Lambda$ such that for every $\lambda \in \Lambda_0$,

$$\Pr\left[E_{1-b} \ \wedge \ \left((P^*, V_{\mathcal{A},b})(1^\lambda) = 1\right)\right] = \mu(\lambda). \tag{10}$$

This, together with Claim 2, implies that for every $\lambda \in \Lambda_0$,

$$\Pr\left[E_b \ \wedge \ \left((P^*, V_{\mathcal{A},b})(1^\lambda) = 1\right)\right] > \frac{1}{p(\lambda)} - \mu(\lambda). \tag{11}$$

Consider the verifier V_b that is identical to the honest verifier V, except that it uses the witness w_b to generate the NIWI, where w_b is the witness corresponding to x^*_b, and similarly to $V_{\mathcal{A},b}$, it does not do the check corresponding to $1-b$, rather only checks the NIWI of the prover and the check corresponding to b. In other words, V_b is identical to $V_{\mathcal{A},b}$ except that he generates x^*_{1-b} honestly (as opposed to $V_{\mathcal{A},b}$ who generates $x^*_{1-b} = (\mathsf{CRS}_{1-b}, \mathsf{pk}_{1-b}, \mathsf{ct}_{1-b})$ where ct_{1-b} is an encryption of 0).

By Claim 1, Equations (10) and (11) imply that there exists a negligible function ν such that for every $\lambda \in \Lambda_0$,

$$\Pr\left[E_{1-b} \ \wedge \ \left((P^*, V_b)(1^\lambda) = 1\right)\right] = \nu(\lambda) \tag{12}$$

and

$$\Pr\left[E_b \ \wedge \ \left((P^*, V_b)(1^\lambda) = 1\right)\right] > \frac{1}{p(\lambda)} - 2\nu(\lambda), \tag{13}$$

where the probabilities are over $b \xleftarrow{\$} \{0,1\}$ and over the randomness of V_b.

We next argue that these two equations contradict the T-security of the NIWI proof system. To this end, we construct a $\mathsf{poly}(T)$-size adversary $\mathcal{M}$ that wins the WI game as described in Definition 5 with non-negligible advantage, as follows.

1. For every $b \in \{0,1\}$ do the following:

   (a) Choose at random $r^1_b \xleftarrow{\$} \{0,1\}^\lambda$ and compute $(\mathsf{CRS}_b, \mathsf{td}_b) = \mathsf{Pol.Setup}(1^\lambda, r^1_b)$.

   (b) Generate $(\mathsf{pk}_b, \mathsf{sk}_b) \leftarrow \mathsf{PKE.Gen}(1^\lambda)$.

   (c) Choose at random $r^2_b \xleftarrow{\$} \{0,1\}^\lambda$ and compute $\mathsf{ct}_b = \mathsf{PKE.Enc}_{\mathsf{pk}_b}(\mathsf{td}_b, r^2_b)$.

   (d) Let $x^*_b = (\mathsf{CRS}_b, \mathsf{pk}_0, \mathsf{ct}_b)$, and let $w_b = (\mathsf{td}_b, r^1_b, r^2_b)$.

2. Choose (x^*_0, x^*_1) to be the instance in the WI game (w.r.t. the NP language L^*_{OR}), and w_0 and w_1 to be the two witnesses.

3. Let π_{NIWI} be the challenge proof generated with respect to witness w_{b^*} for a randomly chosen $b^* \xleftarrow{\$} \{0,1\}$.

4. Compute $(x, \mathsf{msg}) = P^*(\mathsf{pp})$, where $\mathsf{pp} = (x^*_0, x^*_1, \pi_{\mathsf{NIWI}})$.

5. Parse $\mathsf{msg} = \left(\{(x'_b, c'_b, \mathsf{ct}'_b)\}_{b\in\{0,1\}}, c', \pi'_{\mathsf{NIWI}}\right)$.

6. Compute (w', r) such that $c' = \mathsf{Com}(w', r)$.

7. If there exists $b \in \{0,1\}$ such that the following three conditions are satisfied:

   – π'_{NIWI} is accepting,

   – $\pi'_b = \mathsf{PKE.Dec}_{\mathsf{sk}_b}(c'_b)$ satisfies that $\mathsf{Pol.Verify}(\mathsf{CRS}_b, x'_b, \pi'_b) = 1$,

   – $(x'_b, w') \in R_{L'}$,

then output b. Otherwise, output a randomly chosen bit $b \xleftarrow{\$} \{0,1\}$.

We next argue that for every $\lambda \in \Lambda_0$,

$$\Pr[b = b^*] \geq \frac{1}{2} + \frac{1}{3p(\lambda)},$$

contradicting the T-security of the WI property.

To this end, denote by GOOD the event that the following three conditions hold:

- π'_{NIWI} is accepting.
- $\pi'_{b^*} = \mathsf{Dec}_{\mathsf{sk}_{b^*}}(c'_{b^*})$ satisfies $\mathsf{PoI.Verify}(\mathsf{CRS}_{b^*}, x'_{b^*}, \pi'_{b^*}) = 1$.
- There exists b such that $(x'_b, w') \in R_{L'}$.

Equations (12) and (13) imply that for every $\lambda \in \Lambda_0$,

$$\Pr[\mathsf{GOOD}] \geq \frac{1}{p(\lambda)} - \nu(\lambda)$$

and

$$\Pr[\mathsf{GOOD}] - \Pr\left[(b = b^*) \ \wedge \ \mathsf{GOOD}\right] \leq \nu(\lambda).$$

Therefore,

$$\begin{aligned}
&\Pr[b = b^*] = \\
&\Pr\left[(b = b^*) \ \wedge \ \mathsf{GOOD}\right] + \Pr\left[(b = b^*) \ \wedge \ \neg\mathsf{GOOD}\right] \geq \\
&\Pr[\mathsf{GOOD}] - \nu(\lambda) + \frac{1}{2} \cdot \left(1 - \Pr[\mathsf{GOOD}]\right) \geq \\
&\frac{1}{2} + \frac{1}{2} \cdot \Pr[\mathsf{GOOD}] - \nu(\lambda) \geq \\
&\frac{1}{2} + \frac{1}{2p(\lambda)} - 2\nu(\lambda) \geq \\
&\frac{1}{2} + \frac{1}{3p(\lambda)}
\end{aligned}$$

Contradicting the T-security of the NIWI proof system, as desired.

□

Denote by Z the event that both $(P^*, V_{\mathcal{A},b})(1^\lambda) = 1$ and E_{1-b}. By Claim 3, for every $\lambda \in \Lambda$,

$$\Pr[Z] \geq \frac{1}{q(\lambda)}$$

By the definition of $\mathcal{A}$, and the definition of the event E_{1-b}, it holds that

$$\Pr\left[\mathcal{A}(\mathsf{CRS}) = (x', w', \pi') \text{ s.t. } \left((x', w') \in R_{L'}\right) \wedge \left(\mathsf{PoI.Verify}(\mathsf{CRS}, x', \pi') = 1\right) \ \middle| \ Z\right] = 1$$

Thus, we conclude that

$$\Pr\left[\mathcal{A}(\mathsf{CRS}) = (x', w', \pi') \text{ s.t. } \left((x', w') \in R_{L'}\right) \wedge \left(\mathsf{PoI.Verify}(\mathsf{CRS}, x', \pi') = 1\right)\right] \geq \Pr[Z] \geq \frac{1}{q(\lambda)},$$

contradicting the T-security of PoI.

Witness Hiding. Suppose for the sake of contradiction that there exists a poly-size cheating verifier $V^* = (V_1^*, V_2^*)$, a polynomial s, and an infinite set $\Lambda \subseteq \mathbb{N}$, such that for every $\lambda \in \Lambda$,

$$\Pr\left[V_2^*\big(x, V_1^*(1^\lambda, x), \mathsf{msg}_P\big) = w \text{ s.t. } (x, w) \in R_L\right] \geq \frac{1}{s(\lambda)}, \tag{14}$$

where the probability is over $(x, w) \leftarrow \mathcal{D}_\lambda$ and over $\mathsf{msg}_P \leftarrow P(1^\lambda, x, w, \mathsf{pp})$, where $(\mathsf{pp}, \mathsf{st}) = V^*(1^\lambda, x)$, and where pp is the message that V^* sends the prover and st is a secret state that is used by V_2^* to extract w.

Remark 7. *Note that in the description of the protocol, V_2 takes as input $\big(1^\lambda, V_1(1^\lambda), (x, \mathsf{msg}_P)\big)$. In the proof of witness hiding, we change the order to elements, to emphasize that the cheating V_1^* can choose $(\mathsf{pp}, \mathsf{st})$ depending on x. Moreover, for the sake of succinctness, V_2^* does not take 1^λ as input. Rather, we assume (without loss of generality) that st includes 1^λ.*

Remark 8. *We assume without loss of generality that V_1^* always generates an accepting NIWI proof π_{NIWI}. Loosely speaking, this is without loss of generality since P aborts if the NIWI proof π_{NIWI} is rejected, and hence the cheating verifier (V_1^*, V_2^*) does not gain anything by generating a rejecting π_{NIWI}.*

Formally, this is argued as follows: Replace (V_1^*, V_2^*) with the following $(V_{h_1}^*, V_{h_2}^*)$:

- $V_{h_1}^*$*: On input $(1^\lambda, x)$, compute $(\mathsf{pp}, \mathsf{st}) = V_1^*(1^\lambda, x)$. Parse $\mathsf{pp} = (x_0^*, x_1^*, \pi_{\mathsf{NIWI}})$. If*

  $$\mathsf{NIWI.Verify}(x_0^*, x_1^*, \pi_{\mathsf{NIWI}}) = 1$$

  output $\mathsf{pp}_h = \mathsf{pp}$. Else, compute pp_h as the honest verifier and output pp_h.

- $V_{h_2}^*$*: On input $(x, V_{h_1}^*(1^\lambda, x), \mathsf{msg}_P)$, compute $(\mathsf{pp}, \mathsf{st}) = V_1^*(1^\lambda, x)$, and parse $\mathsf{pp} = (x_0^*, x_1^*, \pi_{\mathsf{NIWI}})$. If*

  $$\mathsf{NIWI.Verify}(x_0^*, x_1^*, \pi_{\mathsf{NIWI}}) = 1$$

  output $V_2^*(x, V_1^*(1^\lambda, x), \mathsf{msg}_P)$ where $\mathsf{msg}_P \leftarrow P(1^\lambda, x, w, \mathsf{pp})$. Else, output $V_2^*(x, V_1^*(1^\lambda, x), \bot)$.

The fact that $\mathsf{msg}_P = \bot$ when the NIWI proof of the verifier is rejected implies that the output of $V_{h_2}^*$ is identically distributed to the output of V_2^*. Hence, for every $\lambda \in \Lambda$,

$$\Pr[V_{h_2}^*(x, V_{h_1}^*(1^\lambda, x), \mathsf{msg}_P) \in R_L(x)] = \Pr[V_2^*(x, V_1^*(1^\lambda, x), \mathsf{msg}_P) \in R_L(x)] \geq \frac{1}{s(\lambda)}.$$

where the last inequality holds by Equation (14), and where the probability is over $(x, w) \leftarrow \mathcal{D}_\lambda$ and $\mathsf{msg}_P \leftarrow P(1^\lambda, x, w, \mathsf{pp})$.

Remark 9. *In what follows, we often abuse notation, and denote by $x \leftarrow \mathcal{D}_\lambda$ to denote that x is sampled by sampling $(x, w) \leftarrow \mathcal{D}_\lambda$ and outputting x.*

Subset GOOD: We define the set $\mathsf{GOOD} \subseteq L$, where $x \in \mathsf{GOOD}$ if and only if

$$\Pr[V_2^*\big(x, V_1^*(1^\lambda, x), \mathsf{msg}_P\big) = w \text{ s.t. } (x, w) \in R_L] \geq \frac{1}{2s(\lambda)},$$

where the probability is over $\mathsf{msg}_P \leftarrow P(1^\lambda, x, w, \mathsf{pp})$.

Claim 4. *For every $\lambda \in \Lambda$,*

$$\Pr[x \in \mathsf{GOOD}] \geq \frac{1}{2s(\lambda)}$$

where the probability is over $x \leftarrow \mathcal{D}_\lambda$.

Proof.

$$\begin{aligned}
\Pr[V_2^*(x, V_1^*(1^\lambda, x), \mathsf{msg}_P) \in R_L(x)] = &\Pr[V_2^*(x, V_1^*(1^\lambda, x), \mathsf{msg}_P) \in R_L(x) \mid x \in \mathsf{GOOD}] \cdot \Pr[x \in \mathsf{GOOD}] \\
&+ \Pr[V_2^*(x, V_1^*(1^\lambda, x), \mathsf{msg}_P) \in R_L(x) \mid x \notin \mathsf{GOOD}] \cdot \Pr[x \notin \mathsf{GOOD}] \\
&\leq \Pr[x \in \mathsf{GOOD}] + \frac{1}{2s(\lambda)} \cdot \Pr[x \notin \mathsf{GOOD}] \\
&\leq \Pr[x \in \mathsf{GOOD}] + \frac{1}{2s(\lambda)}
\end{aligned}$$

Hence, for every $\lambda \in \Lambda$, the fact that $\Pr[V_2^*(x, V_1^*(1^\lambda, x), \mathsf{msg}_P) \in R_L(x)] > \frac{1}{s(\lambda)}$ implies that $\Pr[x \in \mathsf{GOOD}] \geq \frac{1}{2s(\lambda)}$, as desired.

□

Trapdoor Set of x: For every x we define the trapdoor set of x, denoted by $\mathsf{td}(x)$, as follows:

$$\mathsf{td}(x) = \{\mathsf{td} : \exists b \in \{0,1\}\ \exists (r_b^1, r_b^2) \text{ s.t. } \big((\mathsf{CRS}_b, \mathsf{td}) = \mathsf{Pol.Setup}(1^\lambda; r_b^1)\big) \wedge \big(\mathsf{ct}_b = \mathsf{Enc}_{\mathsf{pk}_b}(\mathsf{td}, r_b^2)\big)\}$$

where $V_1^*(1^\lambda, x) = (\{(\mathsf{CRS}_b, \mathsf{pk}_b, \mathsf{ct}_b)\}_{b \in \{0,1\}}, \pi_{\mathsf{NIWI}})$. By the perfect soundness of the NIWI,

$$V_{\mathsf{NIWI}}\Big(\{(\mathsf{CRS}_b, \mathsf{pk}_b, \mathsf{ct}_b)\}_{b \in \{0,1\}}, \pi_{\mathsf{NIWI}}\Big) = 1 \implies \mathsf{td}(x) \neq \emptyset.$$

We distinguish between the following two cases:

Case 1. There exists a poly-size computable function f such that for infinitely many $\lambda \in \Lambda$,

$$\Pr[f(x) \in \mathsf{td}(x)] \geq \frac{1}{\mathsf{poly}(\lambda)}, \text{ where } x \leftarrow \mathcal{D}_\lambda | \mathsf{GOOD}. \tag{15}$$

In this case we construct a poly-size $\mathcal{A}$ that given $x \leftarrow \mathcal{D}_\lambda$ outputs a valid witness w with non-negligible probability (breaking the hardness of the language L). $\mathcal{A}$, on input x, does the following:

1. Compute $V_1^*(1^\lambda, x) = \Big(\{(\mathsf{CRS}_b, \mathsf{pk}_b, \mathsf{ct}_b)\}_{b \in \{0,1\}}, \pi_{\mathsf{NIWI}}\Big)$.
2. Compute $f(x) = \mathsf{td}$.

   If td is an invalid trapdoor with respect to both CRS_0 and CRS_1 then abort.
3. Otherwise, td is a valid trapdoor corresponding to CRS_{b^*} for some $b^* \in \{0,1\}$. Namely, there exists $(r_{b^*}^1, r_{b^*}^2)$ for which

$$\big((\mathsf{CRS}_{b^*}, \mathsf{td}) = \mathsf{Pol.Setup}(1^\lambda; r_{b^*}^1)\big) \wedge \big(\mathsf{ct}_{b^*} = \mathsf{PKE.Enc}_{\mathsf{pk}_{b^*}}(\mathsf{td}, r_{b^*}^2)\big).$$

4. Compute $\{(x_b', c_b', \mathsf{ct}_b')\}_{b \in \{0,1\}}$ as the honest prover does. Namely, do the following computations for every $b \in \{0,1\}$:
   - Sample $(x_b', \pi_b') \leftarrow \mathsf{Pol.Gen}(\mathsf{CRS}_b)$,
   - Sample $r_b', s_b' \xleftarrow{\$} \{0,1\}^\lambda$ and compute $c_b' = \mathsf{Com}(\pi_b'; r_b')$ and $\mathsf{ct}_b' = \mathsf{PKE.Enc}_{\mathsf{pk}_b}((\pi_b', r_b'); s_b')$.
5. Compute $w_{b^*} = \mathsf{Pol.Witness}(\mathsf{CRS}_{b^*}, \mathsf{td}, x_{b^*}', \pi_{b^*}')$.
6. Choose at random $u \leftarrow \{0,1\}^{\mathsf{poly}(\lambda)}$, and compute $c' = \mathsf{Com}(w_{b^*}; u)$.
7. Generate a NIWI proof π_{NIWI}' for $(x, x_0', x_1', c') \in L_{\mathsf{Pol}}$, using the witness $(0, b^*, w_{b^*}, u)$.

8. Output $V_2^*\big(x, V_1^*(1^\lambda, x), (\{(x'_b, c'_b, \mathsf{ct}'_b)\}_{b\in\{0,1\}}, c', \pi'_{\mathsf{NIWI}})\big)$

$$\begin{aligned}
&\Pr_{x\leftarrow\mathcal{D}}[\mathcal{A}(x) \in R_L(x)] \geq\\
&\Pr_{x\leftarrow\mathcal{D}}[\mathcal{A}(x) \in R_L(x) \mid x \in \mathsf{GOOD}] \cdot \Pr_{x\leftarrow\mathcal{D}}[x \in \mathsf{GOOD}] \geq\\
&\frac{1}{2s(\lambda)} \cdot \Pr_{x\leftarrow\mathcal{D}}[\mathcal{A}(x) \in R_L(x) \mid x \in \mathsf{GOOD}] \geq\\
&\frac{1}{2s(\lambda)} \cdot \Pr_{x\leftarrow\mathcal{D}}\big[\mathcal{A}(x) \in R_L(x) \mid \big(x \in \mathsf{GOOD}\big) \wedge \big(f(x) \in \mathsf{td}(x)\big)\big] \cdot \Pr_{x\leftarrow\mathcal{D}}[f(x) \in \mathsf{td}(x) \mid x \in \mathsf{GOOD}] \geq\\
&\frac{1}{2s(\lambda)\cdot\mathsf{poly}(\lambda)} \cdot \Pr_{x\leftarrow\mathcal{D}}\big[\mathcal{A}(x) \in R_L(x) \mid \big(x \in \mathsf{GOOD}\big) \wedge \big(f(x) \in \mathsf{td}(x)\big)\big] \geq\\
&\frac{1}{2s(\lambda)\cdot\mathsf{poly}(\lambda)}
\end{aligned}$$

contradicting the hardness of L, where the second inequality follows from Claim 4, the fourth inequality follows from Equation (15) (for infinitely many $\lambda \in \Lambda$), and the last inequality follows from the witness indistinguishability property of the NIWI proof, since $\mathcal{A}(x)$ runs V_2^* on input

$$\big(x, V_1^*(1^\lambda, x), (\{(x'_b, c'_b, \mathsf{ct}'_b)\}_{b\in\{0,1\}}, c', \pi'_{\mathsf{NIWI}})\big),$$

where the message $(\{(x'_b, c'_b, \mathsf{ct}'_b)\}_{b\in\{0,1\}}, c', \pi'_{\mathsf{NIWI}})$ is distributed identically as a message generated by the honest prover, except that the NIWI proof π'_{NIWI} is generated using an alternative witness).

Case 2. For every poly-size computable function f there exists a negligible function μ and for every $\lambda \in \Lambda$

$$\Pr[f(x) \in \mathsf{td}(x)] = \mu(\lambda) \tag{16}$$

where the probability is over $x \leftarrow \mathcal{D}_\lambda|\mathsf{GOOD}$.

Claim 5. *For every $\lambda \in \Lambda$ there exists $b_\lambda \in \{0,1\}$ such that the following holds: For every poly-size adversary $\mathcal{B}$ there exists a negligible function ν such that for every $\lambda \in \Lambda$,*

$$\Pr\Big[\mathcal{B}(x, V_1^*(1^\lambda, x), \mathsf{PKE.Enc}_{\mathsf{pk}_{b_\lambda}}(d)) = d\Big] \leq \frac{1}{2} + \nu(\lambda),$$

where the probability is over $x \leftarrow \mathcal{D}_\lambda|\mathsf{GOOD}$, $d \xleftarrow{\$} \{0,1\}$, and over the randomness of $\mathsf{PKE.Enc}_{\mathsf{pk}_{b_\lambda}}$, where pk_{b_λ} is computed by

$$\Big((\mathsf{CRS}_0, \mathsf{pk}_0, \mathsf{ct}_0), (\mathsf{CRS}_1, \mathsf{pk}_1, \mathsf{ct}_1), \pi_{\mathsf{NIWI}}, \mathsf{st}^*\Big) = V_1^*(1^\lambda, x).$$

Proof. Suppose for contradiction that there exists a poly-size algorithm $\mathcal{B}$, a polynomial q, and an infinite set $\Lambda_0 \subseteq \Lambda$, such that for every $\lambda \in \Lambda_0$ and for every $b \in \{0,1\}$,

$$\Pr\Big[\mathcal{B}\big(x, V_1^*(1^\lambda, x), \mathsf{PKE.Enc}_{\mathsf{pk}_b}(d)\big) = d\Big] \geq \frac{1}{2} + \frac{1}{q(\lambda)} \tag{17}$$

where the probability is over $x \leftarrow \mathcal{D}_\lambda|\mathsf{GOOD}$, $d \xleftarrow{\$} \{0,1\}$, and over the randomness of $\mathsf{PKE.Enc}_{\mathsf{pk}_b}$, where pk_b is computed by

$$\Big((\mathsf{CRS}_0, \mathsf{pk}_0, \mathsf{ct}_0), (\mathsf{CRS}_1, \mathsf{pk}_1, \mathsf{ct}_1), \pi_{\mathsf{NIWI}}\Big) = V_1^*(1^\lambda, x).$$

For every $\lambda \in \Lambda_0$ and every $b \in \{0,1\}$ we define a set $S_{\lambda,b}$ in the image of $\mathcal{D}_\lambda|\mathsf{GOOD}$, where $x \in S_{\lambda,b}$ if and only if

$$\Pr\left[\mathcal{B}\big(x, V_1^*(1^\lambda, x), \mathsf{PKE.Enc}_{\mathsf{pk}_b}(d)\big) = d\right] \geq \frac{1}{2} + \frac{1}{2q(\lambda)} \tag{18}$$

where the probability is over a randomly chosen $d \xleftarrow{\$} \{0,1\}$ and over the randomness of $\mathsf{PKE.Enc}_{\mathsf{pk}_b}$, and where pk_b is computed by

$$\Big((\mathsf{CRS}_0, \mathsf{pk}_0, \mathsf{ct}_0), (\mathsf{CRS}_1, \mathsf{pk}_1, \mathsf{ct}_1), \pi_{\mathsf{NIWI}}\Big) = V_1^*(1^\lambda, x).$$

Note that for every $\lambda \in \Lambda_0$ and every $b \in \{0,1\}$,

$$\begin{aligned}
&\Pr\left[\mathcal{B}\big(x, V_1^*(1^\lambda, x), \mathsf{PKE.Enc}_{\mathsf{pk}_b}(d)\big) = d\right] = \\
&\Pr\left[\mathcal{B}\big(x, V_1^*(1^\lambda, x), \mathsf{PKE.Enc}_{\mathsf{pk}_b}(d)\big) = d \;\middle|\; x \in S_{\lambda,b}\right] \cdot \Pr\left[x \in S_{\lambda,b}\right] + \\
&\Pr\left[\mathcal{B}\big(x, V_1^*(1^\lambda, x), \mathsf{PKE.Enc}_{\mathsf{pk}_b}(d)\big) = d \;\middle|\; x \notin S_{\lambda,b}\right] \cdot \Pr\left[x \notin S_{\lambda,b}\right] \leq \\
&\Pr\left[x \in S_{\lambda,b}\right] + \left(\frac{1}{2} + \frac{1}{2q(\lambda)}\right) \cdot \Pr\left[x \notin S_{\lambda,b}\right] \leq \\
&\Pr\left[x \in S_{\lambda,b}\right] + \left(\frac{1}{2} + \frac{1}{2q(\lambda)}\right).
\end{aligned}$$

This, together with Equation (17), implies that for every $\lambda \in \Lambda_0$ and every $b \in \{0,1\}$,

$$\Pr_{x \leftarrow \mathcal{D}_\lambda|\mathsf{GOOD}}[x \in S_{\lambda,b}] \geq \frac{1}{2q(\lambda)}. \tag{19}$$

By Lemma 3 (in Section 3.2) and by Equation (18), there exists a non-uniform PPT algorithm $\mathcal{E}^*$, and a negligible function μ such that for for every $\lambda \in \Lambda_0$, every $b \in \{0,1\}$, every $x \in S_{\lambda,b}$, and every $m = (m_1, \ldots, m_\lambda) \in \{0,1\}^\lambda$ and $r_1, \ldots, r_\lambda \in \{0,1\}^{\mathsf{poly}(\lambda)}$,

$$\Pr\left[\mathcal{E}^*(x, V_1^*(1^\lambda, x), \mathsf{ct}) = m\right] \geq 1 - \mu(\lambda) \tag{20}$$

where $\mathsf{ct} = \big(\mathsf{PKE.Enc}_{\mathsf{pk}_b}(m_1; r_1), \ldots, \mathsf{PKE.Enc}_{\mathsf{pk}_b}(m_\lambda; r_\lambda)\big)$.

We use $\mathcal{E}^*$ to construct a non-uniform PPT f such that $\Pr[f(x) \in \mathsf{td}(x)]$ is non-negligible, contradicting Equation (16). The function f, on input x in support of $\mathcal{D}_\lambda$, does the following:

- Compute $V_1^*(1^\lambda, x) = \Big(\{(\mathsf{CRS}_b, \mathsf{pk}_b, \mathsf{ct}_b)\}_{b \in \{0,1\}}, \pi_{\mathsf{NIWI}}\Big)$.
- Choose a random $b \xleftarrow{\$} \{0,1\}$ and output $\mathsf{td}'_b = \mathcal{E}^*(x, V_1^*(1^\lambda, x), \mathsf{ct}_b)$.

We next argue that for every $\lambda \in \Lambda_0$,

$$\Pr\Big[f(x) \in \mathsf{td}(x)\Big] \geq \frac{1}{5q(\lambda)},$$

where the probability is over $x \leftarrow \mathcal{D}_\lambda|\mathsf{GOOD}$, contradicting Equation (16).

Let $E_{b,x}$ be the event that for $V_1^*(1^\lambda, x) = \Big((\mathsf{CRS}_0, \mathsf{pk}_0, \mathsf{ct}_0), (\mathsf{CRS}_1, \mathsf{pk}_1, \mathsf{ct}_1), \pi_{\mathsf{NIWI}}\Big)$

$$\exists (r_b^1, r_b^2) \;\text{ s.t. }\; \big((\mathsf{CRS}_b, \mathsf{td}) = \mathsf{Pol.Setup}(1^\lambda; r_b^1)\big) \;\wedge\; \big(\mathsf{ct}_b = \mathsf{PKE.Enc}_{\mathsf{pk}_b}(\mathsf{td}, r_b^2)\big)$$

Since we assumed without loss of generality that π_{NIWI} is always accepting (see Remark 8), by the perfect soundness of the NIWI proof system it holds that

$$\Pr_{x \leftarrow \mathcal{D}_\lambda | \mathsf{GOOD}} \left[E_{0,x} \ \vee \ E_{1,x} \right] = 1. \tag{21}$$

Therefore,

$$\begin{aligned}
&\Pr_{x \leftarrow \mathcal{D}_\lambda | \mathsf{GOOD}} \left[f(x) \in \mathsf{td}(x) \right] = \\
&\frac{1}{2} \cdot \Pr_{x \leftarrow \mathcal{D}_\lambda | \mathsf{GOOD}} \left[\mathcal{E}^*(x, V_1^*(1^\lambda, x), \mathsf{ct}_0) \in \mathsf{td}(x) \right] + \frac{1}{2} \cdot \Pr_{x \leftarrow \mathcal{D}_\lambda | \mathsf{GOOD}} \left[\mathcal{E}^*(x, V_1^*(1^\lambda, x), \mathsf{ct}_1) \in \mathsf{td}(x) \right] \geq \\
&\frac{1}{2} \cdot \Pr_{x \leftarrow \mathcal{D}_\lambda | \mathsf{GOOD}} \left[\mathcal{E}^*(x, V_1^*(1^\lambda, x), \mathsf{ct}_0) \in \mathsf{td}(x) \ \middle| \ x \in S_{\lambda,0} \right] \cdot \Pr[x \in S_{\lambda,0} \Big] + \\
&\frac{1}{2} \Pr_{x \leftarrow \mathcal{D}_\lambda | \mathsf{GOOD}} \left[\mathcal{E}^*(x, V_1^*(1^\lambda, x), \mathsf{ct}_1) \in \mathsf{td}(x) \ \middle| \ x \in S_{\lambda,1} \right] \cdot \Pr \left[x \in S_{\lambda,1} \right] \geq \\
&\frac{1}{4q(\lambda)} \cdot \Pr_{x \leftarrow \mathcal{D}_\lambda | \mathsf{GOOD}} \left[\mathcal{E}^*(x, V_1^*(1^\lambda, x), \mathsf{ct}_0) \in \mathsf{td}(x) \ \middle| \ x \in S_{\lambda,0} \right] + \\
&\frac{1}{4q(\lambda)} \cdot \Pr_{x \leftarrow \mathcal{D}_\lambda | \mathsf{GOOD}} \left[\mathcal{E}^*(x, V_1^*(1^\lambda, x), \mathsf{ct}_1) \in \mathsf{td}(x) \ \middle| \ x \in S_{\lambda,1} \right] \geq \\
&\frac{1}{5q(\lambda)} \left(\Pr_{x \leftarrow \mathcal{D}_\lambda | \mathsf{GOOD}} \left[E_{0,x} \right] + \Pr_{x \leftarrow \mathcal{D}_\lambda | \mathsf{GOOD}} \left[E_{1,x} \right] \right) = \\
&\frac{1}{5q(\lambda)},
\end{aligned}$$

as desired, where the third equation follows from Equation (19), the forth equation follows from Equation (20), and the last equation follows from Equation (21). □

In what follows we construct five non-uniform PPT provers, $P_1^*, P_2^*, P_3^*, P_4^*, P_5^*$, where P_5^* is the honest prover. We argue that for every $i \in [5]$, for every non-uniform PPT adversary $\mathcal{B}$, there exists a negligible function ν such that for every $\lambda \in \Lambda$,

$$\Pr \left[\mathcal{B}\big(x, V_1^*(1^\lambda, x), P_i^*(1^\lambda, x, w, \mathsf{pp}^*)\big) = w \right] \leq \nu(\lambda) \tag{22}$$

where the probability is over $(x, w) \leftarrow \mathcal{D}_\lambda | \mathsf{GOOD}$ and over the random coin tosses of P_i^*, and where $(\mathsf{pp}^*, \mathsf{st}^*) = V_1^*(x)$. Note that this contradicts Equation (14), since Equation (14) implies that:

$$\begin{aligned}
&\frac{1}{s(\lambda)} \leq \\
&\Pr_{x \leftarrow \mathcal{D}}[V_2^*(x, V_1^*(x), P^*(x, V_1^*(x))) \in R_L(x)] = \\
&\Pr_{x \leftarrow \mathcal{D}}[V_2^*(x, V_1^*(x), P^*(x, V_1^*(x))) \in R_L(x) \mid \mathsf{GOOD}] \cdot \Pr[\mathsf{GOOD}] + \\
&\Pr_{x \leftarrow \mathcal{D}}[V_2^*(x, V_1^*(x), P^*(x, V_1^*(x))) \in R_L(x) \mid \neg\mathsf{GOOD}] \cdot \Pr[\neg\mathsf{GOOD}] \leq \\
&\Pr_{x \leftarrow \mathcal{D}}[V_2^*(x, V_1^*(x), P^*(x, V_1^*(x))) \in R_L(x) \mid \mathsf{GOOD}] + \\
&\Pr_{x \leftarrow \mathcal{D}}[V_2^*(x, V_1^*(x), P^*(x, V_1^*(x))) \in R_L(x) \mid \neg\mathsf{GOOD}] \leq \\
&\Pr_{x \leftarrow \mathcal{D}}[V_2^*(x, V_1^*(x), P^*(x, V_1^*(x))) \in R_L(x) \mid \mathsf{GOOD}] + \frac{1}{2s(\lambda)},
\end{aligned}$$

which in turn implies that

$$\Pr_{x \leftarrow \mathcal{D}}[V_2^*(x, V_1^*(x), P^*(x, V_1^*(x))) \in R_L(x) \mid \mathsf{GOOD}] \geq \frac{1}{2s(\lambda)},$$

contradicting Equation (22) for $P_i^* = P_5^* = P$.

Prover P_1^*: We start by defining the non-uniform PPT prover P_1^* that on input $(1^\lambda, x, w, \mathsf{pp}^*)$, where $\mathsf{pp}^* = V_1^*(1^\lambda, x) = (x_0^*, x_1^*, \pi_{\mathsf{NIWI}})$ and where $x_b^* = (\mathsf{CRS}_b, \mathsf{pk}_b, \mathsf{ct}_b)$, ignores the witness w, and does the following:

1. Let $b^* = b_\lambda$, where b_λ is the bit from Claim 5.[13]
2. Compute $(x'_{1-b^*}, \pi'_{1-b^*}) \leftarrow \mathsf{Pol.Gen}(\mathsf{CRS}_{1-b^*})$, choose at random $r'_{1-b^*}, s'_{1-b^*} \leftarrow \{0,1\}^\lambda$ and compute $c'_{1-b^*} = \mathsf{Com}(\pi'_{1-b^*}; r'_{1-b^*})$ and $\mathsf{ct}'_{1-b^*} = \mathsf{PKE.Enc}_{\mathsf{pk}_{1-b^*}}((\pi'_{1-b^*}, r'_{1-b^*}); s'_{1-b^*})$.
3. Generate a pair $(x'_{b^*}, w'_{b^*}) \in R_{L'}$ such that x'_{b^*} is distributed according to $\mathcal{D}'$.
4. Generate $c'_{b^*} \leftarrow \mathsf{Com}(0)$ and $\mathsf{ct}'_{b^*} \leftarrow \mathsf{PKE.Enc}_{\mathsf{pk}_{b^*}}(0)$.
5. Choose at random $v' \leftarrow \{0,1\}^\lambda$ and compute $c' = \mathsf{Com}(w'_{b^*}; v')$.
6. Generate a NIWI proof for π'_{NIWI} for $(x, x'_0, x'_1, c') \in L_{\mathsf{Pol}}$, using witness (b^*, w'_{b^*}, v').
7. Output $\big(\{(x'_b, c'_b, \mathsf{ct}'_b)\}_{b\in\{0,1\}}, c', \pi'_{\mathsf{NIWI}}\big)$.

Claim 6. *For every non-uniform* PPT *adversary $\mathcal{B}$, there exists a negligible function ν such that for every $\lambda \in \Lambda$,*

$$\Pr\Big[\mathcal{B}\big(x, V_1^*(1^\lambda, x), P_1^*(1^\lambda, x, w, \mathsf{pp}^*)\big) = w\Big] \le \nu(\lambda)$$

and

$$\Pr\Big[\mathcal{B}\Big(x, V_1^*(1^\lambda, x), P_1^*(1^\lambda, x, w, \mathsf{pp}^*), \mathsf{PKE.Enc}_{\mathsf{pk}_{b^*}}(d)\Big) = d\Big] \le \frac{1}{2} + \nu(\lambda),$$

where the probabilities are over $x \leftarrow \mathcal{D}_\lambda|\mathsf{GOOD}$, $d \xleftarrow{\$} \{0,1\}$, and over the randomness of P_1^*, where $(\mathsf{pp}^*, \mathsf{st}^*) = V_1^*(1^\lambda, x)$. In addition, the second probability is also over $\mathsf{PKE.Enc}_{\mathsf{pk}_{b^*}}$ *where*

$$\mathsf{pp}^* = \Big((\mathsf{CRS}_0, \mathsf{pk}_0, \mathsf{ct}_0), (\mathsf{CRS}_1, \mathsf{pk}_1, \mathsf{ct}_1), \pi_{\mathsf{NIWI}}\Big).$$

Proof. The first equation follows from the fact that the messages of V_1^* and P_1^* are efficiently computable given only $(1^\lambda, x)$. The second equation follows from Claim 5, together with the fact that P_1^* is efficiently computable given only $(1^\lambda, x)$. □

Prover P_2^*: We next define a non-uniform PPT algorithm P_2^*, which is identical to P_1^* except that P_2^* uses the witness w corresponding to x in the NIWI proof.

Claim 7. *For every non-uniform* PPT *adversary $\mathcal{B}$, there exists a negligible function ν such that for every $\lambda \in \Lambda$,*

$$\Pr\Big[\mathcal{B}\big(x, V_1^*(1^\lambda, x), P_2^*(1^\lambda, x, w, \mathsf{pp}^*)\big) = w\Big] \le \nu(\lambda)$$

and

$$\Pr\Big[\mathcal{B}\Big(x, V_1^*(1^\lambda, x), P_2^*(1^\lambda, x, w, \mathsf{pp}^*), \mathsf{PKE.Enc}_{\mathsf{pk}_{b^*}}(d)\Big) = d\Big] \le \frac{1}{2} + \nu(\lambda),$$

where the probability is over $(x, w) \leftarrow \mathcal{D}_\lambda|\mathsf{GOOD}$ and over the random coin tosses of P_2^*, and where $(\mathsf{pp}^*, \mathsf{st}^*) = V_1^*(x)$. In addition, the second probability is also over $\mathsf{PKE.Enc}_{\mathsf{pk}_{b^*}}$ *where*

$$\mathsf{pp}^* = \Big((\mathsf{CRS}_0, \mathsf{pk}_0, \mathsf{ct}_0), (\mathsf{CRS}_1, \mathsf{pk}_1, \mathsf{ct}_1), \pi_{\mathsf{NIWI}}\Big).$$

[13] P_1^* has the bit b_λ hard-wired into it.

Claim 7 follows from Claim 6, together with the security property of the NIWI.

Prover P_3^*: We next define a non-uniform PPT algorithm P_3^*, which is identical to P_2^* except that P_3^* generates $c' \leftarrow \mathsf{Com}(0)$ as opposed to $c' \leftarrow \mathsf{Com}(w'_{b^*})$. In more detail, P_3^* on input $(1^\lambda, x, w, \mathsf{pp}^*)$ does the following:

1. For every $b \in \{0,1\}$, compute $(x'_b, \pi'_b) \leftarrow \mathsf{Pol.Gen}(\mathsf{CRS}_b)$.
2. Choose at random $r'_{1-b^*}, s'_{1-b^*} \leftarrow \{0,1\}^\lambda$ and compute $c'_{1-b^*} = \mathsf{Com}(\pi'_{1-b^*}; r'_{1-b^*})$ and $\mathsf{ct}'_{1-b^*} = \mathsf{PKE.Enc}_{\mathsf{pk}_{1-b^*}}((\pi'_{1-b^*}, r'_{1-b^*}); s'_{1-b^*})$.
3. Generate $c'_{b^*} \leftarrow \mathsf{Com}(0)$ and $\mathsf{ct}'_{b^*} \leftarrow \mathsf{PKE.Enc}_{\mathsf{pk}_{b^*}}(0)$.
4. Generate $c' \leftarrow \mathsf{Com}(0)$.
5. Generate a NIWI proof for π'_{NIWI} for $(x, x'_0, x'_1, c') \in L_{\mathsf{Pol}}$, using the witness w.
6. Output $\left(\{(x'_b, c'_b, \mathsf{ct}'_b)\}_{b \in \{0,1\}}, c', \pi'_{\mathsf{NIWI}}\right)$.

Claim 8. *For every non-uniform* PPT *adversary $\mathcal{B}$, there exists a negligible function ν such that for every $\lambda \in \Lambda$,*

$$\Pr\left[\mathcal{B}\big(x, V_1^*(1^\lambda, x), P_3^*(1^\lambda, x, w, \mathsf{pp}^*)\big) = w\right] \leq \nu(\lambda)$$

and

$$\Pr\left[\mathcal{B}\Big(x, V_1^*(1^\lambda, x), P_3^*(1^\lambda, x, w, \mathsf{pp}^*), \mathsf{PKE.Enc}_{\mathsf{pk}_{b^*}}(d)\Big) = d\right] \leq \frac{1}{2} + \nu(\lambda),$$

where the probability is over $(x, w) \leftarrow \mathcal{D}_\lambda|\mathsf{GOOD}$ and over the random coin tosses of P_3^*, and where $(\mathsf{pp}^*, \mathsf{st}^*) = V_1^*(x)$. In addition, the second probability is also over $\mathsf{PKE.Enc}_{\mathsf{pk}_{b^*}}$ where

$$\mathsf{pp}^* = \Big((\mathsf{CRS}_0, \mathsf{pk}_0, \mathsf{ct}_0), (\mathsf{CRS}_1, \mathsf{pk}_1, \mathsf{ct}_1), \pi_{\mathsf{NIWI}}\Big).$$

Claim 8 follows from Claim 7, together with the hiding property of the commitment scheme.

Prover P_4^*: We next define a non-uniform PPT algorithm P_4^*, which is identical to P_3^* except that P_4^* generates $c'_{b^*} \leftarrow \mathsf{Com}(\pi'_{b^*})$ as opposed to $c'_{b^*} \leftarrow \mathsf{Com}(0)$. In more detail, P_4^* on input $(1^\lambda, x, w, \mathsf{pp}^*)$ does the following:

1. For every $b \in \{0,1\}$, compute $(x'_b, \pi'_b) \leftarrow \mathsf{Pol.Gen}(\mathsf{CRS}_b)$.
2. Choose at random $r'_{1-b^*}, s'_{1-b^*} \leftarrow \{0,1\}^\lambda$ and compute $c'_{1-b^*} = \mathsf{Com}(\pi'_{1-b^*}; r'_{1-b^*})$ and $\mathsf{ct}'_{1-b^*} = \mathsf{PKE.Enc}_{\mathsf{pk}_{1-b^*}}((\pi'_{1-b^*}, r'_{1-b^*}); s'_{1-b^*})$.
3. Generate $c'_{b^*} \leftarrow \mathsf{Com}(\pi'_{b^*})$ and $\mathsf{ct}'_{b^*} \leftarrow \mathsf{PKE.Enc}_{\mathsf{pk}_{b^*}}(0)$.
4. Generate $c' \leftarrow \mathsf{Com}(0)$.
5. Generate a NIWI proof for π'_{NIWI} for $(x, x'_0, x'_1, c') \in L_{\mathsf{Pol}}$, using the witness w.
6. Output $\left(\{(x'_b, c'_b, \mathsf{ct}'_b)\}_{b \in \{0,1\}}, c', \pi'_{\mathsf{NIWI}}\right)$.

Claim 9. *For every non-uniform* PPT *adversary $\mathcal{B}$, there exists a negligible function ν such that for every $\lambda \in \Lambda$,*

$$\Pr\left[\mathcal{B}\big(x, V_1^*(1^\lambda, x), P_4^*(1^\lambda, x, w, \mathsf{pp}^*)\big) = w\right] \leq \nu(\lambda)$$

and

$$\Pr\left[\mathcal{B}\Big(x, V_1^*(1^\lambda, x), P_4^*(1^\lambda, x, w, \mathsf{pp}^*), \mathsf{PKE.Enc}_{\mathsf{pk}_{b^*}}(d)\Big) = d\right] \leq \frac{1}{2} + \nu(\lambda),$$

where the probability is over $(x, w) \leftarrow \mathcal{D}_\lambda|\mathsf{GOOD}$ *and over the random coin tosses of* P_4^**, and where* $(\mathsf{pp}^*, \mathsf{st}^*) = V_1^*(x)$*. In addition, the second probability is also over* $\mathsf{PKE.Enc}_{\mathsf{pk}_{b^*}}$ *where*

$$\mathsf{pp}^* = \Big((\mathsf{CRS}_0, \mathsf{pk}_0, \mathsf{ct}_0), (\mathsf{CRS}_1, \mathsf{pk}_1, \mathsf{ct}_1), \pi_{\mathsf{NIWI}}\Big).$$

Claim 9 follows from Claim 8, together with the hiding property of the commitment scheme.

Prover P_5^*: Finally, we define P_5^* to be the honest prover. Note that the only difference between P_5^* and P_4^* is in the way ct_{b^*} is generated: P_5^* generates ct'_{b^*} as $\mathsf{ct}'_{b^*} \leftarrow \mathsf{PKE.Enc}_{\mathsf{pk}_{b^*}}(\pi'_{b^*}, r'_{b^*})$ (where r'_{b^*} was the randomness used to commit to π'_{b^*}), whereas P_4^* generates it as $\mathsf{ct}'_{b^*} \leftarrow \mathsf{PKE.Enc}_{\mathsf{pk}_{b^*}}(0)$.

Claim 10. *For every non-uniform* PPT *adversary* $\mathcal{B}$*, there exists a negligible function* ν *such that for every* $\lambda \in \Lambda$*,*

$$\Pr\Big[\mathcal{B}\big(x, V_1^*(1^\lambda, x), P_5^*(1^\lambda, x, w, \mathsf{pp}^*)\big) = w\Big] \leq \nu(\lambda)$$

where the probability is over $(x, w) \leftarrow \mathcal{D}_\lambda|\mathsf{GOOD}$ *and over the random coin tosses of* P_5^**, and where* $(\mathsf{pp}^*, \mathsf{st}^*) = V_1^*(x)$*.*

Proof. Claim 9, together with the KDM security of the underlying encryption scheme, implies that

$$\Big(x, V_1^*(1^\lambda, x), P_4^*(1^\lambda, x, w, \mathsf{pp}^*), \mathsf{PKE.Enc}_{\mathsf{pk}_{b^*}}(0)\Big) \approx \Big(x, V_1^*(1^\lambda, x), P_4^*(1^\lambda, x, w, \mathsf{pp}^*), \mathsf{PKE.Enc}_{\mathsf{pk}_{b^*}}(\pi'_{b^*}, r'_{b^*})\Big) \tag{23}$$

Suppose for the sake of contradiction that there exists a non-uniform PPT adversary $\mathcal{B}$ and a polynomial p such that for infinitely many $\lambda \in \Lambda$,

$$\Pr\Big[\mathcal{B}\big(x, V_1^*(1^\lambda, x), P_5^*(1^\lambda, x, w, \mathsf{pp}^*)\big) = w\Big] \geq \frac{1}{p(\lambda)} \tag{24}$$

where the probability is over $(x, w) \leftarrow \mathcal{D}_\lambda|\mathsf{GOOD}$ and over the random coin tosses of P_5^*, and where $(\mathsf{pp}^*, \mathsf{st}^*) = V_1^*(x)$. We use $\mathcal{B}$ to construct a non-uniform PPT adversary $\mathcal{A}$ that contradicts Equation (23). Algorithm $\mathcal{A}$, on input $\Big(x, V_1^*(1^\lambda, x), P_4^*(1^\lambda, x, w, \mathsf{pp}^*), \mathsf{ct}\Big)$, runs $\mathcal{B}$ on input $\big(x, V_1^*(1^\lambda, x), P_4^*(1^\lambda, x, w, \mathsf{pp}^*)\big)$, while replacing the encryption ct'_{b^*} generated by P_4^* with the ciphertext ct. If $\mathcal{B}$ outputs w then $\mathcal{A}$ outputs 1 and otherwise $\mathcal{A}$ outputs a random guess $b \xleftarrow{\$} \{0, 1\}$. Note that if $\mathsf{ct} \leftarrow \mathsf{PKE.Enc}_{\mathsf{pk}^*}(0)$ then the input fed to $\mathcal{B}$ is generated identically to $P_4^*(1^\lambda, x, w, \mathsf{pp}^*)$, whereas if $\mathsf{ct} \leftarrow \mathsf{PKE.Enc}_{\mathsf{pk}^*}(\pi_{b^*}, r'_{b^*})$ then the input fed to $\mathcal{B}$ is generated identically to $P_5^*(1^\lambda, x, w, \mathsf{pp}^*)$. Therefore, by Equation (24) for infinitely many $\lambda \in \Lambda$,

$$\begin{aligned}
&\Pr\Big[\mathcal{A}\Big(x, V_1^*(1^\lambda, x), P_4^*(1^\lambda, x, w, \mathsf{pp}^*), \mathsf{PKE.Enc}_{\mathsf{pk}_{b^*}}(\pi_{b^*}, r'_{b^*})\Big) = 1\Big] - \\
&\Pr\Big[\mathcal{A}\Big(x, V_1^*(1^\lambda, x), P_4^*(1^\lambda, x, w, \mathsf{pp}^*), \mathsf{PKE.Enc}_{\mathsf{pk}_{b^*}}(0)\Big) = 0\Big] \geq \\
&\frac{1}{p(\lambda)} + \left(1 - \frac{1}{p(\lambda)}\right)\frac{1}{2} - \frac{1}{2} - \mathsf{negl}(\lambda) \geq \frac{1}{3p(\lambda)},
\end{aligned}$$

contradicting Equation (23), where the first inequality follows from the definition of $\mathcal{B}$ together with Equation (24) and Claim 9. □

□

7 Acknowledgements

Authors would like to thank Omer Paneth for collaboration at an early stage of this project.

References

[BBK$^+$16] Nir Bitansky, Zvika Brakerski, Yael Kalai, Omer Paneth, and Vinod Vaikuntanathan. 3-message zero knowledge against human ignorance. In *Theory of Cryptography Conference*, pages 57–83. Springer, 2016.

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

[BCKP17] Nir Bitansky, Ran Canetti, Yael Tauman Kalai, and Omer Paneth. On virtual grey box obfuscation for general circuits. *Algorithmica*, 79(4):1014–1051, 2017.

[BCPR16] Nir Bitansky, Ran Canetti, Omer Paneth, and Alon Rosen. On the existence of extractable one-way functions. *SIAM Journal on Computing*, 45(5):1910–1952, 2016.

[BGI$^+$01] Boaz Barak, Oded Goldreich, Russell Impagliazzo, Steven Rudich, Amit Sahai, Salil Vadhan, and Ke Yang. On the (im)possibility of obfuscating programs. Cryptology ePrint Archive, Report 2001/069, 2001. `http://eprint.iacr.org/`.

[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*, pages 671–684. ACM, 2018.

[BOV05] Boaz Barak, Shien Jin Ong, and Salil P. Vadhan. Derandomization in cryptography. *IACR Cryptology ePrint Archive*, 2005:365, 2005.

[BP02] Mihir Bellare and Adriana Palacio. Gq and schnorr identification schemes: Proofs of security against impersonation under active and concurrent attacks. In *Annual International Cryptology Conference*, pages 162–177. Springer, 2002.

[BP04] Mihir Bellare and Adriana Palacio. The knowledge-of-exponent assumptions and 3-round zero-knowledge protocols. In *Annual International Cryptology Conference*, pages 273–289. Springer, 2004.

[BP12] Nir Bitansky and Omer Paneth. Point obfuscation and 3-round zero-knowledge. In *Theory of Cryptography Conference*, pages 190–208. Springer, 2012.

[BR93] Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *Proceedings of the 1st ACM conference on Computer and communications security*, pages 62–73. ACM, 1993.

[CCRR18] Ran Canetti, Yilei Chen, Leonid Reyzin, and Ron D. Rothblum. Fiat-shamir and correlation intractability from strong kdm-secure encryption. 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 I*, pages 91–122, 2018.

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

[DN00] Cynthia Dwork and Moni Naor. Zaps and their applications. In *Foundations of Computer Science, 2000. Proceedings. 41st Annual Symposium on*, pages 283–293. IEEE, 2000.

[DSYC17] Yi Deng, Xuyang Song, Jingyue Yu, and Yu Chen. On instance compression, schnorr/guillou-quisquater, and the security of classic protocols for unique witness relations. *IACR Cryptology ePrint Archive*, 2017:390, 2017.

[FFS88] Uriel Feige, Amos Fiat, and Adi Shamir. Zero-knowledge proofs of identity. *Journal of Cryptology*, 1(2):77–94, 1988.

[FS90] Uriel Feige and Adi Shamir. Witness indistinguishable and witness hiding protocols. In *Proceedings of the twenty-second annual ACM symposium on Theory of computing*, pages 416–426. ACM, 1990.

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

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

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

[GOS06] Jens Groth, Rafail Ostrovsky, and Amit Sahai. Non-interactive zaps and new techniques for nizk. In *CRYPTO*, volume 4117, pages 97–111. Springer, 2006.

[HRS09] Iftach Haitner, Alon Rosen, and Ronen Shaltiel. On the (im) possibility of arthur-merlin witness hiding protocols. In *Theory of Cryptography Conference*, pages 220–237. Springer, 2009.

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

[JKKR17] Abhishek Jain, Yael Tauman Kalai, Dakshita Khurana, and Ron Rothblum. Distinguisher-dependent simulation in two rounds and its applications. In *Annual International Cryptology Conference*, pages 158–189. Springer, 2017.

[Pas03] Rafael Pass. Simulation in quasi-polynomial time, and its application to protocol composition. In *International Conference on the Theory and Applications of Cryptographic Techniques*, pages 160–176. Springer, 2003.