A Transform for NIZK Almost as Efficient and General as the Fiat-Shamir Transform Without Programmable Random Oracles

Michele Ciampi[1], Giuseppe Persiano[2], Luisa Siniscalchi[1], and Ivan Visconti[1]

[1] DIEM, University of Salerno, ITALY
{mciampi,lsiniscalchi,visconti}@unisa.it
[2] DISA-MIS, University of Salerno, ITALY
giuper@gmail.com

Abstract. The Fiat-Shamir (FS) transform is a popular technique for obtaining practical zero-knowledge argument systems. The FS transform uses a hash function to generate, without any further overhead, non-interactive zero-knowledge (NIZK) argument systems from public-coin honest-verifier zero-knowledge (public-coin HVZK) proof systems. In the proof of zero knowledge, the hash function is modeled as a *programmable* random oracle (PRO).

In TCC 2015, Lindell embarked on the challenging task of obtaining a similar transform with improved heuristic security. Lindell showed that, for several interesting and practical languages, there exists an efficient transform in the *non-programmable* random oracle (NPRO) model that also uses a common reference string (CRS). A major contribution of Lindell's transform is that zero knowledge is proved without random oracles and this is an important step towards achieving efficient NIZK arguments in the CRS model without random oracles.

In this work, we analyze the efficiency and generality of Lindell's transform and notice a significant gap when compared with the FS transform. We then propose a new transform that aims at filling this gap. Indeed our transform is almost as efficient as the FS transform and can be applied to a broad class of public-coin HVZK proof systems. Our transform requires a CRS and an NPRO in the proof of soundness, similarly to Lindell's transform.

1 Introduction

Non-interactive zero-knowledge (NIZK) proofs[1] introduced in [24, 6, 5] are widely used in Cryptography. Such proofs allow a prover to convince a verifier with just one message about the membership of an instance x in a language L without leaking any additional information. NIZK proofs are not possible without a setup

[1] When discussing informally we will use the word proof to mean both an unconditionally sound proof and a computationally sound proof (i.e., an argument). Only in the more formal part of the paper we will make a distinction between arguments and proofs.

assumption and the one proposed initially in [5] is the existence of a *Common Reference String* (CRS) received as input both by the prover and the verifier. The CRS model has been the standard setup for NIZK in the last 25 years. Another setup that has been proposed in literature is the existence of registered public keys in [2, 21, 13].

Starting with the breakthrough of [29, 30] we know that NIZK proofs in the CRS model exist for any NP language with the additional appealing feature of using just one CRS for any polynomial number of proofs. Moreover NIZK proofs and their stronger variations [48, 23, 39] have been shown to be not only interesting for their original goal of being a non-interactive version of classic zero-knowledge (ZK) proofs [36, 37], but also because they are powerful building blocks in many applications (e.g., for CCA encryption [45], ZAPs [27, 28]).

Efficient NIZK. Generic constructions of NIZK proofs are rather inefficient since they require to first compute an NP reduction and then to apply the NIZK proof for a given NP-complete language to the instance output by the reduction. A significant progress in efficiency has been proposed in [40] where several techniques have been proposed to obtain efficient NIZK proofs that can be used in bilinear groups.

The most popular use of NIZK proofs in real-world scenarios consists in taking an efficient *interactive* public-coin honest-verifier zero-knowledge (HVZK) proof system and in making it a NIZK argument through the so called *Fiat-Shamir (FS) transform* [31]. The FS transform replaces the verifier by calls to a hash function on input the transcript so far. In the random oracle [3] (RO) model the hash function can only be evaluated through calls to an oracle that answers as a random function. The security proof allows the simulator for HVZK to program the RO (i.e., the simulator decides how to answer to a query) and this allows to convert the entire transcript of a public-coin HVZK proof into a single message that is indistinguishable from the single message computed by a honest NIZK prover. The efficiency of the FS transform led to many practical applications. The transform is also a method to obtain signatures of knowledge, as discussed in [14].

The main disadvantage of the FS transform is the fact that the random oracle methodology has been proved to be unsound both in general [7] and both for the specific case [35, 4] of turning identification schemes into signatures as considered in [31]. Nevertheless, the examples of constructions proved secure in the RO model and insecure for any concrete hash function are seemingly artificial while no natural construction has been successfully attacked yet. Therefore the RO methodology remains widely used in practice.

The FS transform applied to 3-round HVZK proofs is one of the major uses of the RO model for real-world protocols, therefore any progress in this research direction (either on the security of the transform, or on its efficiency, or on its generality) is of extreme interest.

In [38] Groth showed an efficient transform for NIZK where soundness is proved requiring a programmable RO while no random oracle is needed to prove zero knowledge.

Efficient NIZK with designated/registered verifiers. A first attempt to get efficient NIZK arguments from some restricted class of 3-round public-coin HVZK proofs without ROs was done by [21] (the proof of soundness required complexity leveraging) and later on by [13] that achieved a weaker form of soundness in the registered public-key model. The limitation of this model is that a NIZK proof can be verified only by a designated verifier (i.e., the proof requires a secret known to the verifier). Moreover there is an inconvenient preliminary registration phase where the verifier has to register her public key.

Lindell's transform. Very recently, in [43], Lindell proposed a very interesting transform that can be seen as an attempt towards obtaining efficient constructions without random oracles. Starting from a Σ-protocol for a language L (i.e., a special type of 3-round public-coin HVZK proof used already in several efficient constructions of zero knowledge [19, 44, 25, 51, 10, 54, 1, 46, 49]), Lindell shows how to construct an efficient NIZK[2] argument system for L in the CRS model. Two are the major advantages of Lindell's transform with respect to the FS transform. First, in Lindell's transform the proof of ZK does not need the existence of a random oracle and this allows to avoid some issues due to protocol composition [52]. We remark that the proof of ZK for Lindell's transform needs a CRS but this is unavoidable as one-round ZK in the plain model is possible only for trivial languages. Second, the soundness of Lindell's transform can be proved by relying on a *non-programmable random oracle* (NPRO). An NPRO is a RO that in the protocol and in the security proofs can be used only as a black box and can not be programmed by a simulator or by the adversary of a reduction. This is a considerable advantage compared to the FS transform since replacing a RO by an NPRO is a step towards removing completely the need of ROs in a cryptographic construction. Indeed the work of Lindell goes precisely in the direction of solving a major open problem in Cryptography: obtaining an efficient RO-free transform for NIZK arguments to be used in place of the FS transform.

The main drawback of Lindell's transform is that it requires extra computation on top of the one needed to run the Σ-protocol for the language L. In contrast, the FS transform does not incur into any overhead on top of a 3-round public-coin HVZK proof for L. In addition, since 3-round public-coin HVZK proofs are potentially less demanding than Σ-protocols, we have that requiring a Σ-protocol as starting protocol for a transform instead of a public-coin HVZK proof may already result in an efficiency loss.

Lindell's transform is based on a primitive named *dual-mode* (DM) commitment scheme (DMCS). A DMCS is based on a membership-hard language Λ and each specific commitment takes as input an instance ρ of Λ and has the following property: if $\rho \notin \Lambda$, the DM commitment is perfectly binding; on the other hand, if $\rho \in \Lambda$, the DM commitment can be arbitrarily equivocated if a

[2] Lindell's NIZK argument is a not an argument of knowledge in contrast to the NIZK argument obtained through an FS transform.

witness for $\rho \in \Lambda$ is known. Moreover, the two modes are indistinguishable[3]. Lindell showed that DMCSs can be constructed efficiently from Σ-protocols for membership-hard languages and also provided a concrete example based on the language of Diffie-Hellman tuples (DH). Then, Lindell's transform shows how to combine DM commitments and Σ-protocols along with a hash function[4] to obtain an efficient NIZK argument.

1.1 Our Results

In this paper, we continue the study of generic and efficient transforms from 3-round public-coin HVZK proofs to NIZK arguments.

We start by studying the generality and efficiency of Lindell's transform in terms of the Σ-protocol used for instantiating the DMCS (and in turn instantiating the CRS) and the Σ-protocol to which the transform is applied. As a result, we point out a significant gap in generality and efficiency of Lindell's transform compared to the FS transform.

Then we show an improved transform that is based on weaker requirements. Specifically, our transform only requires computational HVZK and optimal soundness instead of perfect special HVZK[5] and special soundness. More interestingly and surprisingly despite being based on weaker requirements, our transform is also significantly more efficient than Lindell's transform and very close to the efficiency of the FS transform. We next discuss our contributions in more details.

The classes of Σ-protocols needed in [43]. Lindell defines Σ-protocols as 3-round public-coin proofs that enjoy *perfect* special HVZK and special soundness. The former property means that the simulator on input any valid statement x and challenge e can compute (a, z) such that the triple (a, e, z) is perfectly indistinguishable from an accepting transcript where the verifier sends e as challenge. Special soundness instead means that from any two accepting transcripts (a, e, z) and (a, e', z') for the same statement x that share the first message but have different challenges $e \neq e'$, one can efficiently compute a witness w for $x \in L$. Lindell in [42] shows a construction of a DMCS from any (defined as above) Σ-protocol for a membership-hard language [6].

The efficiency of Lindell's transform. Lindell's transform uses a DMCS derived from a Σ-protocol $\Pi_\Lambda = (\mathcal{P}_\Lambda, \mathcal{V}_\Lambda)$ for language Λ whose commitment algorithm `com` works by running the simulator of Π_Λ. The CRS contains an instance ρ of Λ along with the description of a hash function h. The argument produced by

[3] A similar notion was introduced in [11, 12] and a scheme with similar features was proposed in [22].

[4] In the proof of soundness this function will be modeled as an NPRO.

[5] The latest version of Lindell's transform [42] works by assuming just perfect special HVZK instead of *strong* perfect special HVZK needed in [43].

[6] The construction in [43] needs an additional property that however is enjoyed by classic Σ-protocols as we discuss in App. A.

the NIZK $\Pi = (\mathcal{P}, \mathcal{V})$ for $x \in L$ starting from a Σ-protocol $\Pi_L = (\mathcal{P}_L, \mathcal{V}_L)$ for L is computed as a tuple (a', e, z, r) where $a' = \mathtt{com}(a, r)$, $e = h(x|a')$, and z is the 3rd round of Π_L answering to the challenge e and having a as first round. The verifier checks that a' is a commitment of a with randomness r, that e is the output of $h(x|a')$ and that (a, e, z) is accepted by $\mathcal{V}_L$.

As an example, in [43] Lindell discussed the use of the Σ-protocol for the language DH for which the transform produces a very efficient NIZK proof; indeed the additional cost is of only 8 modular exponentiations: 4 to be executed by the prover and 4 by the verifier.

In this work we notice however that there is a caveat when analyzing the efficiency of Lindell's transform. The caveat is due to the message space of the DMCS. Indeed, once the CRS is fixed the max length of a message that can be committed to with only one execution of com is limited to the challenge length l_Λ of Π_Λ. Therefore in case the first round a of Π_L is much longer than l_Λ, the transform of Lindell requires multiple executions of com therefore suffering of a clear efficiency loss.

We show indeed in Tables 2 and 3 that Lindell's transform can generate in the resulting NIZK argument a blow up of the computations compared to what $\mathcal{P}_L$ and $\mathcal{V}_L$ actually do, and therefore compared to the FS transform.

Our Transform. In this paper, we present a different transform that is closer to the FS transform both on generality and on efficiency.

Our transform can be used to obtain a NIZK for any language L with a 3-round HVZK proofs enjoying optimal soundness (i.e., a weaker soundness requirement compared to special soundness). The CRS can be instantiated based on any membership-hard language Λ with a 3-round HVZK proofs enjoying optimal soundness. More specifically, we do not require perfect HVZK nor special HVZK for the involved Σ-protocols. Moreover, instead of special soundness, we will just require that, for any false statement and any first round message a, there is at most one challenge c that can be answered correctly. This is clearly a weaker requirement than special soundness and was already used by [44].

Essentially we just need that both protocols Π_L and Π_Λ are 3-round public-coin HVZK proofs with optimal soundness. Our transform produces a NIZK argument $\Pi = (\mathcal{P}, \mathcal{V})$ that does not require multiple executions of Π_L and Π_Λ and, therefore, it remains efficient under any scenario without suffering of the previously discussed issue about challenge spaces in Lindell's transform.

Techniques. We start by considering the FS transform in the NPRO model and by noticing that, as already claimed and proved in [53], if the original 3-round public-coin HVZK proof is witness indistinguishable (WI)[7], then the transformed protocol is still WI, and of course the proof of WI is RO free.

Notice that as in [43], $\mathcal{P}$ and $\mathcal{V}$ need a common hash function (modeled as an NPRO in the soundness proof) to run the protocol and this can be enforced through a setup (i.e., a non-programmable CRS [47], or a global hash

[7] We use WI both to mean witness indistinguishable and witness indistinguishability.

function [9]). The use of the FS transform in the NPRO model is not sufficient for our purposes. Indeed we want generality and the HVZK proof might not be witness indistinguishable. Moreover we should make a witness available to the simulator. We solve this problem by using the OR composition of 3-round perfect HVZK proofs proposed in [18]. We will let the prover $\mathcal{P}$ for NIZK to prove that either $x \in L \vee \rho \in \Lambda$. We notice that in [18] the proposed OR composition is proved to guarantee WI only when applied to two instances of the same language having a public-coin *perfect* HVZK proof. We can avoid this limitation using a generalization discussed already in [33, 32] that allows the OR composition different protocols for different languages relying on *computational HVZK* only.

1.2 Comparison

Here we compare the computational effort, both for the prover and the verifier, required to execute Lindell's NIZK argument, our NIZK argument and the FS one. The properties of the three transforms are summarized in Table 1. The cost for the prover can be found in Table 2, while the one for the verifier can be found in Table 3. The comparison of the computational effort is performed with respect to three Σ-protocols[8]. Roughly speaking, in the comparisons, we consider the CRS to contain an instance of the the language DH of Diffie-Hellman triples with respect to 1024-bit prime p_{CRS} and consider two Σ-protocols: the one to prove that a triples is Diffie-Hellman[9] with respect to a prime p, for which we consider the cases in which p is 1024-bit and 2048-bit long[10], and the Σ-protocol for graph isomorphism (GI). For the Σ-protocol for graph isomorphism, we count only the modular exponentiations and do not count other operations (e.g., random selection of a permutation and generation of the adjacency matrix of permuted graphs) since they are extremely efficient and clearly dominated by the cost of modular exponentiations. A detailed description of the Σ-protocols and of the way we measure the computational effort is found in Section 6.

The tables give evidence of the fact that while Lindell's transform on some specific cases can replace the FS transform by paying a small overhead, in other cases there is a significant loss in performance. Our transform instead remains very close to the FS transform both when considering the amount of computation and when considering the generality of the protocols that can be given as input to the transform.

[8] We consider the same Σ-protocol discussed in [43] and in addition we consider the one for Graph Isomorphism since it has the special property of having a very long first round that can be computed very efficiently.

[9] See Section 6 for a formal definition of the polynomial relation and the respective Σ-protocols.

[10] Clearly, in case p is such that $|p| < |p_{\text{CRS}}|$, then Lindell's transform has a slightly smaller number of exponentiations with respect to the number of exponentiations that we count in the tables.

Transform	$HVZK$ for Λ	$HVZK$ for L	Soundness	Model
Lindell [42]	special + perfect	special + perfect	special	NPRO+CRS
This paper	computational	computational	optimal	NPRO+CRS
FS	/	computational	classic	PRO

Table 1: Requirements for the proofs in input to the three transforms.

	DH		GI
Transform	$\lvert p\rvert = 1024$	$\lvert p\rvert = 2048$	n vertices
Lindell [42]	$2 \mod p + 12 \mod p_{\text{CRS}}$	$2 \mod p + 20 \mod p_{\text{CRS}}$	$4n^2 \mod p_{\text{CRS}}$
This paper	$2 \mod p + 4 \mod p_{\text{CRS}}$	$2 \mod p + 4 \mod p_{\text{CRS}}$	$4 \mod p_{\text{CRS}}$
FS	$2 \mod p$	$2 \mod p$	/

Table 2: Efficiency of the three transforms: modular exponentiations for the prover.

	DH		GI
Transform	$\lvert p\rvert = 1024$	$\lvert p\rvert = 2048$	n vertices
Lindell [42]	$4 \mod p + 12 \mod p_{\text{CRS}}$	$4 \mod p + 20 \mod p_{\text{CRS}}$	$4n^2 \mod p_{\text{CRS}}$
This paper	$4 \mod p + 4 \mod p_{\text{CRS}}$	$4 \mod p + 4 \mod p_{\text{CRS}}$	$4 \mod p_{\text{CRS}}$
FS	$4 \mod p$	$4 \mod p$	/

Table 3: Efficiency of the three transforms: modular exponentiations for the verifier.

Which protocols can be given in input to the transform? We stress that our transform allows for additional proof systems to be used for instantiating the CRS and for obtaining a NIZK argument system. This is not only a theoretical progress. Indeed there exist efficient constructions such as the one of [51] that is a variation of the one of [44]. The construction of [51] is an efficient 3-round HVZK proof system with optimal soundness for a language L and is not a Σ-protocol for the corresponding relation $\mathcal{R}_L$. For further details, see App. B.

2 HVZK Proof Systems and Σ-Protocols

We denote the security parameter by n and use "$|$" as concatenation operator (i.e., if a and b are two strings then by $a|b$ we denote the concatenation of a and b). For a finite set S, $x \leftarrow S$ denotes the algorithm that chooses x from S with uniform distribution.

A polynomial-time relation $\mathcal{R}$ (or *polynomial relation*, in short) is a subset of $\{0,1\}^* \times \{0,1\}^*$ such that membership of (x,w) in $\mathcal{R}$ can be decided in time polynomial in $|x|$. For $(x,w) \in \mathcal{R}$, we call x the *instance* and w a *witness* for x. For a polynomial-time relation $\mathcal{R}$, we define the $\mathtt{NP}$-language $L_{\mathcal{R}}$ as $L_{\mathcal{R}} = \{x|\exists w : (x,w) \in \mathcal{R}\}$. We will model a random oracle as a random function $\mathcal{O} : \{0,1\}^* \rightarrow \{0,1\}^n$. Analogously, unless otherwise specified, for an $\mathtt{NP}$-language

L we denote by $\mathcal{R}_L$ the corresponding polynomial-time relation (that is, $\mathcal{R}_L$ is such that $L = L_{\mathcal{R}_L}$).

We remark that for simplicity we will omit the modulus in modular arithmetic calculations.

For two interactive machines A and B, we denote by $\langle A(\alpha), B(\beta)\rangle(\gamma)$ the distribution of B's output after running on private input β with A using private input α, both running on common input γ. Typically, one of the two machines receives the security parameter 1^n as input.

Definition 1. *A pair of PPT interactive machines $(\mathcal{P}_L, \mathcal{V}_L)$ constitutes a* proof system *(resp., an* argument system*) for* NP*-language L, if the following conditions hold:*

- *Completeness. For every $x \in L$ and w such that $(x, w) \in \mathcal{R}_L$, it holds:*

$$\text{Prob}\,[\ \langle \mathcal{P}_L(w, 1^n), \mathcal{V}_L\rangle(x) = 1\ ] = 1.$$

- *Soundness. For every interactive (resp., PPT interactive) machine $\mathcal{P}^\star_L$, there exists a negligible function ν such that for every $x \notin L$ and every z:*

$$\text{Prob}\,[\ \langle \mathcal{P}^\star_L(z, 1^n), \mathcal{V}_L\rangle(x) = 1\ ] \leq \nu(n).$$

An interactive protocol $\Pi_L = (\mathcal{P}_L, \mathcal{V}_L)$ is *public coin* if, at every round, $\mathcal{V}_L$ simply tosses a predetermined number of coins (random challenge) and sends the outcome to the prover.

In a 3-round public-coin protocol $\Pi_L = (\mathcal{P}_L, \mathcal{V}_L)$ for an NP-language L, $\mathcal{P}_L$ and $\mathcal{V}_L$ receive the common input x and, additionally, $\mathcal{P}_L$ receives security parameter 1^n in unary and w such that $(x, w) \in \mathcal{R}_L$ as private input. The interaction, with challenge length l, proceeds as follows:

The 3-round public-coin protocol Π_L:

1. $\mathcal{P}_L$, on input $1^n, x$ and w, computes message a and sends it to $\mathcal{V}_L$.
2. $\mathcal{V}_L$ chooses a random challenge $e \leftarrow \{0,1\}^l$ and sends it to $\mathcal{P}_L$.
3. $\mathcal{P}_L$, on input x, w, e, and the randomness used to compute a, computes message z and sends it to $\mathcal{V}_L$.
4. $\mathcal{V}_L$ decides to accept or reject based on its view (i.e., (x, a, e, z)).

A triple (a, e, z) of messages exchanged during the execution of a 3-round proof (resp., argument) system is called a *3-round transcript.* We say that a 3-round transcript (a, e, z) is an *accepting transcript* for x if the argument system Π_L instructs $\mathcal{V}_L$ to accept based on the values (x, a, e, z). Two accepting 3-rounds transcripts (a, e, z) and (a', e', z') for an instance x constitute a *collision* if $a = a'$ and $e \neq e'$.

Definition 2. *A* 3*-round proof or argument system $\Pi_L = (\mathcal{P}_L, \mathcal{V}_L)$ for* NP*-language L is* Honest-Verifier Zero Knowledge (HVZK) *if there exists a PPT simulator algorithm* Sim *that takes as input security parameter 1^n and instance*

$x \in L$ and outputs an accepting transcript for x. Moreover, the distribution of the output of the simulator on input x is computationally indistinguishable from the distribution of the honest transcript obtained when $\mathcal{V}_L$ and $\mathcal{P}_L$ run Π_L on common input x and any private input w such that $(x, w) \in \mathcal{R}_L$.

If the transcripts are identically distributed we say that Π_L is perfect HVZK.

Definition 3. *A 3-round public-coin proof system $\Pi_L = (\mathcal{P}_L, \mathcal{V}_L)$ for language L with challenge length l enjoys* optimal soundness *if for every $x \notin L$ and for every first-round message a there is at most one challenge $e \in \{0,1\}^l$ for which there exists a third-round message z such that (a, e, z) is accepting for x.*

Note that any 3-round public-coin optimally sound proof system with challenge length l has soundness error 2^{-l} [44].

Definition 4. *A 3-round public-coin proof system $\Pi_L = (\mathcal{P}_L, \mathcal{V}_L)$ with challenge length l is a Σ-protocol for an* NP*-language L if it enjoys the following properties:*

- *Completeness. If $(x, w) \in \mathcal{R}_L$ then all honest 3-round transcripts for (x, w) are accepting.*
- *Special Soundness. There exists an efficient algorithm* Extract *that, on input x and a collision for x, outputs a witness w such that $(x, w) \in \mathcal{R}_L$.*
- *Special Honest Verifier Zero Knowledge (special HVZK). There exists a PPT simulator algorithm* Sim *that takes as input security parameter 1^n, $x \in L$ and $e \in \{0,1\}^l$ and outputs an accepting transcript for x where e is the challenge. Moreover for all l-bit strings e, the distribution of the output of the simulator on input (x, e) is perfect indistinguishable from the distribution of the 3-round honest transcript obtained when $\mathcal{V}_L$ sends e as challenge and $\mathcal{P}_L$ runs on common input x and any private input w such that $(x, w) \in \mathcal{R}_L$.*

Sometimes, we will abuse notion and say that a proof system or Σ-protocol is for a polynomial relation $\mathcal{R}$ instead of referring to NP-language $L_{\mathcal{R}}$.

It is easy to see that Σ-protocols enjoy optimal soundness. The converse, however, is not true. See Appendix B for an example of an optimal-sound 3-round public-coin proof system that does not enjoy special soundness (and is special perfect HVZK).

In order not to overburden the descriptions of protocols and simulators, we will omit the specification of the security parameter when it is clear from the context.

2.1 3-Round Public-Coin HVZK Proofs and WI

Following [33], for an NP-language L, we define $\hat{L}$ to be the input language that includes both L and all false instances that are well formed and can be used by an adversarial prover in order to prove a false statement. More formally, $L \subseteq \hat{L}$ and membership in $\hat{L}$ can be tested in polynomial time. We implicitly assume that a verifier executes the protocol only if the common input $x \in \hat{L}$; otherwise, it rejects immediately.

Definition 5. *A 3-round public-coin proof system $\Pi = (\mathcal{P}_L, \mathcal{V}_L)$ is* Witness Indistinguishable (WI) *for polynomial relation $\mathcal{R}$ if, for every malicious verifier $\mathcal{V}_L^\star$, there exists a negligible function ν such that for all x, w, w' with $(x, w) \in \mathcal{R}$ and $(x, w') \in \mathcal{R}$, it holds that:*

$$|\mathrm{Prob}\,[\ \langle \mathcal{P}_L(w, 1^n), \mathcal{V}_L^\star \rangle(x) = 1\ ] - \mathrm{Prob}\,[\ \langle \mathcal{P}_L(w', 1^n), \mathcal{V}_L^\star \rangle(x) = 1\ ]| \leq \nu(n).$$

The notion of a perfect *WI* 3*-round proof system is obtained by requiring that $\nu(n) = 0$.*

Sometimes we abuse the above definition and say that a proof system is WI for a NP-language L instead of referring to the associated polynomial relation $\mathcal{R}_L$.

We recall the following result.

Theorem 1 ([18]). *Every 3-round public-coin proof system with perfect HVZK for an* NP*-language L is perfect WI for $\mathcal{R}_L$.*

2.2 Challenge Lengths of 3-Round HVZK Proofs

Challenge-length amplification. The challenge of a 3-round public-coin proof system with HVZK and optimal soundness can be extended through parallel repetition.

Lemma 1. *Let Π_L be a 3-round public-coin proof system with optimal soundness for* NP*-language L that enjoys perfect HVZK and has challenge length l. The protocol Π_L^k consisting of k parallel instances of Π_L is a 3-round public-coin proof system for relation L that enjoys perfect HVZK, has optimal soundness and has challenge length $k \cdot l$.*

Proof. The HVZK it is preserved by Π_L^k for the same arguments of [18]. About the optimal soundness of Π_L^k, it is simple to see that if the protocol Π_L^k in not optimal sound then also Π_L is not optimal sound.

A similar lemma can be proved for a Σ-protocol (as in [32, 15, 16]) for which HVZK is not perfect.

Challenge-length reduction. We now show that starting from any 3-round public-coin proof system that enjoys HVZK and has optimal soundness with challenge length l, one can construct a 3-round public-coin proof system that still enjoys HVZK, has optimal soundness but works with a shorter challenge. Moreover perfect HVZK is preserved. A similar transformation was shown in [20] for the case of Σ-protocol that are special perfect HVZK.

Lemma 2. *Let Π_L be a HVZK 3-round public-coin proof system for L with optimal soundness and challenge length l. Then for every $l' < l$, there exists a 3-round public-coin proof system Π_L' for L with HVZK and optimal soundness and challenge length l'. Protocol Π_L' has the same efficiency as Π_L and, moreover, if Π_L is perfect HVZK so is Π_L'.*

Proof. Following is a description of Π'_L.
Common input: instance x for an $\mathtt{NP}$-language L.
Private input of $\mathcal{P}'_L$: w s.t. $(x, w) \in \mathcal{R}_L$.
The protocol Π'_L:

1. $\mathcal{P}'_L$ computes $a \leftarrow \mathcal{P}_L(x, w)$ and sends it to $\mathcal{V}'_L$;
2. $\mathcal{V}'_L$ randomly chooses challenge $e \leftarrow \{0,1\}^{l'}$ and sends it to $\mathcal{P}'_L$;
3. $\mathcal{P}'_L$ randomly chooses $pad \leftarrow \{0,1\}^{(l-l')}$, sets $e' = e|pad$, computes $z \leftarrow \mathcal{P}_L(x, w, a, e')$ and sends $z' = (z, pad)$ to $\mathcal{V}'_L$;
4. $\mathcal{V}'_L$ outputs the output of $\mathcal{V}_L(x, a, e|pad, z)$.

Completeness follows directly from the completeness of Π.

HVZK. We can consider the simulator Sim', that on input x runs as follows:

1. run $(a, e', z) \leftarrow \mathsf{Sim}(x)$;
2. set pad equal to the last $l - l'$ bits of e', and set e equal to the fist l' bits of e';
3. output $(a, e, (z, pad))$.

This concludes the proof.

Optimal soundness follows directly from the optimal soundness of Π.
The following theorem follows from Lemma 1 and 2,

Theorem 2. *Suppose* $\mathtt{NP}$*-language* L *admits a HVZK 3-round public-coin proof system* Π_L *that has optimal soundness and challenge length* l*. Then for any* $l' > 0$ *there exists HVZK 3-round public-coin proof system* Π'_L *that has optimal soundness and challenge length* l'*. If* $l' \leq l$ *then* $\Pi^{'}_L$ *is as efficient as* Π_L*. Otherwise the communication and computation complexities of* $\Pi^{'}_L$ *are at most* l'/l *times the ones of* Π_L*. Moreover, perfect HVZK is preserved.*

2.3 3-Round Public-Coin HVZK Proofs for OR Composition of Statements

In this section we recall the construction of [18] that starts from a HVZK 3-round public-coin proof system Π_L for an $\mathtt{NP}$-language L and constructs a HVZK 3-round public-coin proof system $\Pi_{L \vee L}$ for the "OR" language of L; that is the $\mathtt{NP}$-language $L \vee L = \{(x_0, x_1) : x_0 \in L \vee x_1 \in L\}$. Below we give the descriptions of the prover $\mathcal{P}_{L \vee L}$ and of the verifier $\mathcal{V}_{L \vee L}$ of $\Pi_{L \vee L}$. In the description, we let Sim denote the simulator for Π_L and l denote the challenge length of Π_L. We also let $b \in \{0,1\}$ be such that w is a witness for $x_b \in L$; that is, $(x_b, w) \in \mathcal{R}_L$.

Common input: instances x_0, x_1 for an $\mathtt{NP}$-language L.
Private input of $\mathcal{P}_{L \vee L}$: w s.t $(x_0, x_1, w) \in \hat{\mathcal{R}}_{L \vee L}$.
The protocol $\Pi_{L \vee L}$:

1. $\mathcal{P}_{L \vee L}$ computes $a_b \leftarrow \mathcal{P}_L(x_b, w)$, $(a_{1-b}, e_{1-b}, z_{1-b}) \leftarrow \mathsf{Sim}(x_{1-b})$ and sends (a_0, a_1) to $\mathcal{V}_{L \vee L}$.

2. $\mathcal{V}_{L\vee L}$ chooses at random challenge $e \leftarrow \{0,1\}^l$ and sends e to $\mathcal{P}_{L\vee L}$.
3. $\mathcal{P}_{L\vee L}$ sets $e_b = e \oplus e_{1-b}$, computes $z_b \leftarrow \mathcal{P}_L(x_b, w, a_b, e_b)$ and outputs $\Big((e_0, e_1), (z_0, z_1)\Big)$.
4. $\mathcal{V}_{L\vee L}\Big((x_0, x_1), (a_0, a_1), e, ((e_0, e_1), (z_0, z_1))\Big)$. $\mathcal{V}_{L\vee L}$ accepts if and only if $e = e_0 \oplus e_1$ and $\mathcal{V}_L(x_0, a_0, e_0, z_0) = 1$ and $\mathcal{V}_L(x_1, a_1, e_1, z_1) = 1$.

Theorem 3 ([18, 33]). *If Π_L is a HVZK 3-round public-coin proof system with optimal soundness for* NP*-language L then $\Pi_{L\vee L}$ is a HVZK 3-round public-coin proof system with optimal soundness for* NP*-language $L \vee L$ and is WI for polynomial-time relation*

$$\mathcal{R}_{L\vee L} = \Big\{((x_0, x_1), w) : \Big((x_0, w) \in \mathcal{R}_L \wedge x_1 \in L\Big) \vee \Big((x_1, w) \in \mathcal{R}_L \wedge x_0 \in L\Big)\Big\}.$$

Moreover if Π_L is perfect HVZK then $\Pi_{L\vee L}$ is perfect WI for polynomial-time relation

$$\hat{\mathcal{R}}_{L\vee L} = \Big\{((x_0, x_1), w) : \Big((x_0, w) \in \mathcal{R}_L \wedge x_1 \in \hat{L}\Big) \vee \Big((x_1, w) \in \mathcal{R}_L \wedge x_0 \in \hat{L}\Big)\Big\}.$$

We remark that results of [18, 33] are known to hold for Σ-protocols, but in the proof of WI they use only HVZK. Therefore their results also hold starting from a HVZK 3-round public-coin proof system with optimal soundness (and not necessarily special soundness) that we consider in the above theorem. Indeed we observe that $\Pi_{L\vee L}$ has optimal soundness for the following reason. Suppose that $\Pi_{L\vee L}$ does not enjoy optimal soundness. This means that for a false instance and the same first round (a_0, a_1) there are two accepting conversation, namely:

$$\Big((a_0, a_1), e, ((e_0, e_1), (z_0, z_1))\Big), \Big((a_0, a_1), e', ((e'_0, e'_1), (z'_0, z'_1))\Big)$$

with $e \neq e'$. Then it must be the case that for some $b = 0$ or $b = 1$, $e_b \neq e'_b$ and then (a_b, e_b, z_b) (a_b, e'_b, z'_b) are two accepting transcripts with the same first round for the protocol Π_L, and thus the optimal soundness of Π_L is violated.

It is possible to extend the above construction to handle two different NP-languages L_0, L_1 that admit HVZK 3-round public-coin proof system with optimal soundness. Indeed by Theorem 2, we can assume, without loss of generality, that L_0 and L_1 have 3-round public-coin proof systems Π_{L_0} and Π_{L_1} with the same challenge length. Assuming that L_0 and L_1 have 3-round public-coin proof systems Π_{L_0} and Π_{L_1} that are HVZK and have optimal soundness with the same challenge length. We can apply the same construction outlined above to obtain a 3-round public-coin proof system $\Pi_{L_0\vee L_1}$ that enjoys HVZK and has optimal soundness for relation

$$\hat{\mathcal{R}}_{L_0\vee L_1} = \Big\{((x_0, x_1), w) : \Big((x_0, w) \in \mathcal{R}_{L_0} \wedge x_1 \in \hat{L}_1\Big) \vee \Big((x_1, w) \in \mathcal{R}_{L_1} \wedge x_0 \in \hat{L}_0\Big)\Big\}.$$

We have the following theorem.

Theorem 4. *If Π_{L_0} and Π_{L_1} are HVZK 3-round public-coin proof systems with optimal soundness for* NP*-languages L_0 and L_1 then $\Pi_{L_0 \vee L_1}$ is a HVZK 3-round public-coin proof system with optimal soundness for the for* NP*-language $L_0 \vee L_1 = \{(x_0, x_1) : x_0 \in L_0 \vee x_1 \in L_1\}$ and is WI for polynomial-time relation*

$$\mathcal{R}_{L_0 \vee L_1} = \Big\{((x_0, x_1), w) : \Big((x_0, w) \in \mathcal{R}_{L_0} \wedge x_1 \in L_1\Big) \vee \Big((x_1, w) \in \mathcal{R}_{L_1} \wedge x_0 \in L_0\Big)\Big\}.$$

Moreover, if Π_{L_0} and Π_{L_1} are perfect then $\Pi_{L_0 \vee L_1}$ is perfect WI for polynomial-time relation $\hat{\mathcal{R}}_{L \vee L}$.

3 Non-Interactive Argument Systems

Part of the definitions of this section are taken from [43].

Definition 6. *A* non-interactive argument system *for an* NP*-language L consists of three PPT machines $(\mathcal{CRS}, \mathcal{P}, \mathcal{V})$, that have the following properties:*

- *Completeness: for all $(x, w) \in \mathcal{R}_L$, it holds that:*
$$\text{Prob}\left[\ \sigma \leftarrow \mathcal{CRS}(1^n); \mathcal{V}(\sigma, x, \mathcal{P}(\sigma, x, w)) = 1\ \right] = 1.$$
- *Adaptive Soundness: for every PPT function $f : \{0,1\}^{poly(n)} \to \{0,1\}^n \setminus L$ for all PPT prover $\mathcal{P}^\star$, there exists a negligible function ν, such that for all n:*
$$\text{Prob}\left[\ \sigma \leftarrow \mathcal{CRS}(1^n); \mathcal{V}^{\mathcal{O}}(\sigma, f(\sigma), \mathcal{P}^{\star\mathcal{O}}(\sigma)) = 1\ \right] \leq \nu(n)$$
where $\mathcal{O} : \{0,1\}^* \to \{0,1\}^n$ is a random function.

Definition 7. *A non-interactive argument system is* adaptive unbounded zero knowledge *(NIZK) for an* NP*-language L if there exists a probabilistic PPT simulator S such that for every PPT function*

$$f : \{0,1\}^{\texttt{poly(n)}} \to \Big(\{0,1\}^n \times \{0,1\}^{\texttt{poly(n)}}\Big) \cap \mathcal{R}_L,$$

for every polynomial $p(\cdot)$ and for every PPT malicious verifier $\mathcal{V}^\star$, there exists a negligible function ν such that,

$$\left|\text{Prob}\left[\ \mathcal{V}^\star\left(R_f(\mathcal{P}^f(n,p))\right) = 1\ \right] - \text{Prob}\left[\ \mathcal{V}^\star\left(S_f(n,p)\right) = 1\ \right]\right| \leq \nu(n)$$

where f_1 and f_2 denote the first and second output of f, respectively, and $R_f(\mathcal{P}^f(n,p))$ and $S_f(n,p)$ denote the output from the following experiments.

***Real proofs** $R_f(\mathcal{P}^f(n,p))$:*

- *$\sigma \leftarrow \mathcal{CRS}(1^n)$ a common reference string is sampled.*
- *For $i = 1, \ldots, p(n)$ (initially $\boldsymbol{x}$ and $\boldsymbol{\pi}$ are empty):*
  - *$x_i \leftarrow f_1(\sigma, \boldsymbol{x}, \boldsymbol{\pi})$: the next statement x_i to be proven is chosen.*
  - *$\pi_i \leftarrow \mathcal{P}(\sigma, f_1(\sigma, \boldsymbol{x}, \boldsymbol{\pi}), f_2(\sigma, \boldsymbol{x}, \boldsymbol{\pi}))$: the ith proof is generated.*
  - *set $\boldsymbol{x} = x_1 \ldots x_i$ and $\boldsymbol{\pi} = \pi_1 \ldots \pi_i$.*

- *output* $(\sigma, \boldsymbol{x}, \boldsymbol{\pi})$.

Simulation $S_f(n, p)$:

- $\sigma \leftarrow S(1^n)$ *a common reference string is sampled.*
- *For* $i = 1, \ldots, p(n)$ *(initially* $\boldsymbol{x}$ *and* $\boldsymbol{\pi}$ *are empty):*
  - $x_i \leftarrow f_1(\sigma, \boldsymbol{x}, \boldsymbol{\pi})$: *the next statement* x_i *to be proven is chosen.*
  - $\pi_i \leftarrow S(x_i)$: *simulator* S *generates a simulated proof* π_i *that* $x_i \in L$.
  - *set* $\boldsymbol{x} = x_1 \ldots x_i$ *and* $\boldsymbol{\pi} = \pi_1 \ldots \pi_i$.
- *output* $(\sigma, \boldsymbol{x}, \boldsymbol{\pi})$.

Definition 8. *A non-interactive argument system is* adaptive unbounded witness indistinguishable *(NIWI) for an* `NP`*-language* L *if for every PPT adversary* $\mathcal{V}^\star$, *for every PPT function*

$$f : \{0,1\}^{\mathtt{poly(n)}} \rightarrow \Big(\{0,1\}^n \times \{0,1\}^{\mathtt{poly(n)}} \times \{0,1\}^{\mathtt{poly(n)}}\Big) \cap \mathcal{R}_L^\wedge,$$

and for every polynomial $p(\cdot)$, *there exists a negligible function* ν *such that*

$$\Big|\mathrm{Prob}\Big[\ \mathcal{V}^\star(R_0^{\mathcal{P},f}(n,p)) = 1\ \Big] - \mathrm{Prob}\Big[\ \mathcal{V}^\star(R_1^{\mathcal{P},f}(n,p)) = 1\ \Big]\Big| \leq \nu(n),$$

where $\mathcal{R}_L^\wedge = \{(x, w^0, w^1) : (x, w^0) \in \mathcal{R}_L \wedge (x, w^1) \in \mathcal{R}_L\}$ *and* $R_b^{\mathcal{P},f}$ *is the following experiment.* $R_b^{\mathcal{P},f}(n,p)$:

- $\sigma \leftarrow \mathcal{CRS}(1^n)$.
- *For* $i = 1, \ldots, p(n)$ *(initially* $\boldsymbol{x}$ *and* $\boldsymbol{\pi}$ *are empty):*
  - $(x_i, w_i^0, w_i^1) \leftarrow f(\sigma, \boldsymbol{x}, \boldsymbol{\pi})$:
    statement x_i *to be proven and witnesses* w_i^0, w_i^1 *for* x_i *are generated.*
  - $\pi_i \leftarrow \mathcal{P}(\sigma, x_i, w_i^b)$: *the ith proof is generated.*
  - *set* $\boldsymbol{x} = x_1 \ldots x_i$ *and* $\boldsymbol{\pi} = \pi_1 \ldots \pi_i$.
- *output* $(\sigma, \boldsymbol{x}, \boldsymbol{\pi})$.

4 NIWI Argument Systems from 3-Round HVZK Proofs

In this section we discuss the FS transform in the NPRO model in order to obtain a NIWI argument system $\Pi = (\mathcal{P}, \mathcal{V})$ for a polynomial relation $\mathcal{R}_L$. We start from a 3-round public-coin WI HVZK proof system with optimal soundness $\Pi_L = (\mathcal{P}_L, \mathcal{V}_L)$ for L. $\mathcal{P}$ and $\mathcal{V}$ have access to an NPRO $H : \{0,1\}^* \rightarrow \{0,1\}^n$. We describe Π below and we assume that the challenge length of Π_L is the security parameter n.

Common input: instance x for `NP`-language L.
Private input to $\mathcal{P}$: w s.t. $(x, w) \in \mathcal{R}_L$.
Common reference string: $\mathcal{CRS}$ samples a key s for a hash function family H and sets $\sigma = s$.

1. $\mathcal{P} \rightarrow \mathcal{V}$: The prover $\mathcal{P}$ executes the following steps:

1.1. $a \leftarrow \mathcal{P}_L(x, w)$;
1.2. $e \leftarrow H_s(x, a)$;
1.3. $z \leftarrow \mathcal{P}_L(x, w, a, e)$;
1.4. send $\pi = (a, e, z)$ to $\mathcal{V}$.
2. $\mathcal{V}'$s output: $\mathcal{V}$ outputs 1 if and only if $\mathcal{V}_L(x, a, e, z) = 1$ and $e = H_s(x, a)$.

The following theorem was proved by Yung and Zhao in [53] (see Claim 1, page 4). For completeness, we provide a proof of the claim below.

Theorem 5 ([53]). *Let Π_L be a 3-round public-coin WI proof system for the polynomial relation $\mathcal{R}_L$. Then Π is adaptive WI for $\mathcal{R}_L$ in the CRS model.*

Proof. We show that Π is adaptive WI for $\mathcal{R}_L$ through the following hybrids.

1. $\mathcal{H}_1$ is the experiment $R_0^{\mathcal{P},f}(n, p)$ (Definition 8), where $\mathcal{P}$ for $j = 1, \ldots, p(n)$ executes Π and outputs π_j using the first of the two witnesses given in output by f.
2. $\mathcal{H}_i$ (with $i > 0$) differs from $\mathcal{H}_1$ in the first i interactions, where $\mathcal{P}$ executes Π using the second witness given in output by f. Namely: $\mathcal{P}$ on input (x_j, w_j^1) executes Π and outputs π_j using w_j^1 for all $j : 1 \leq j < i$. Instead, for the interactions $i \leq j < p(n) + 1$, $\mathcal{P}$ on input (x_j, w_j^0) executes Π using w_j^0 as a witness and outputs π_j.
3. $\mathcal{H}_{p(n)+1}$ is the experiment $R_1^{\mathcal{P},f}(n, p)$ (Definition 8), where $\mathcal{P}$ for $j = 1, \ldots, p(n)$ executes Π and outputs π_j using the second witness given in output by f.

$\mathcal{H}_i \approx \mathcal{H}_{i+1}$: Suppose there exists a malicious adversary $\mathcal{V}^\star$ that distinguishes between the experiments $\mathcal{H}_i$ and $\mathcal{H}_{i+1}$ with $1 \leq i \leq p(n)$, then we can show that there exists an adversary $\mathcal{A}$ that breaks the WI property of Π_L. The reduction works as follows.

1. For $j = 1, \ldots, i - 1$, $\mathcal{A}$ on input (x_j, w_j^1) executes Π using w_j^1 to obtain π_j.
2. For $j = i$, $\mathcal{A}$ interacts with the WI challenger of Π_L as follows:
   (a) $\mathcal{A}$ has on input (x_j, w_j^0, w_j^1) and sends it to the challenger of WI;
   (b) the challenger computes and sends the first message a_j to $\mathcal{A}$;
   (c) $\mathcal{A}$ computes $e_j = H_s(a_j)$ and sends it to the challenger of WI;
   (d) the challenger computes and sends z_j to $\mathcal{A}$;
   (e) $\mathcal{A}$ sends $\pi_j = (a_j, e_j, z_j)$ to $\mathcal{V}^\star$;
   (f) $\mathcal{A}$ adds to $\boldsymbol{x}$ the theorem x_j and to $\boldsymbol{\pi}$ the proof π_j.
3. $\forall j = i + 1, \ldots, p(n)$ $\mathcal{A}$ on input (x_j, w_j^0) executes Π using w_j^0 to obtain π_j.
4. Set $\boldsymbol{x} = x_1, \ldots, x_{p(n)}$ and $\boldsymbol{\pi} = \pi_1, \ldots, \pi_{p(n)}$.

$\mathcal{A}$ sends $\boldsymbol{x}$ and $\boldsymbol{\pi}$ to $\mathcal{V}^\star$ and outputs what $\mathcal{V}^\star$ outputs.

We now observe that if the challenger of WI has used the first witness we are in $\mathcal{H}_i$ otherwise we are in $\mathcal{H}_{i+i}$. It follows that $R_0^{\mathcal{P},f}(n, p) \equiv \mathcal{H}_1 \approx \cdots \approx \mathcal{H}_{p(n)} \approx \mathcal{H}_{p(n)+1} \equiv R_1^{\mathcal{P},f}(n, p)$ to conclude the proof.

Adaptive soundness. To prove soundness we follow [43] and use the fact that, for every function g, with a sufficiently large co-domain, relation $\mathcal{R} = \{(x, g(x))\}$ is evasive [8] in the NPRO model. A relation $\mathcal{R}$ is *evasive* if, given access to a random oracle $\mathcal{O}$, it is infeasible to find a string x so that the pair $(x, \mathcal{O}(x)) \in \mathcal{R}$.

Theorem 6. *Let Π_L be a 3-round public-coin proof system with optimal soundness for the* $\mathtt{NP}$*-language L, and let H be a non programmable random oracle. Then, Π is a non-interactive argument system with (adaptive) soundness for L in the NPRO model.*

Proof. Completeness of Π follows from the completeness of Π_L. Let $\mathcal{O}$ be an NPRO. In order to prove the soundness of Π we use the fact that for any function g, the relation $\mathcal{R} = \{(x, g(x))\}$ is evasive. We define the function g s.t. $g(x, a) = e$, where there exists z such that the transcript (a, e, z) is accepting for the instance x. If $x \notin L$ by the optimal soundness property we have that for every a there is a single e for which there is some z so that (a, e, z) is accepting. Therefore g is a function, as required and it follows that the relation $\mathcal{R} = \{((x, a), g(x, a))\}$ is evasive. Suppose that there exist a polynomial function f and a malicious prover $\mathcal{P}^\star$ such that $\mathcal{P}^\star$ proves a false statement (i.e., $\mathcal{V}^{\mathcal{O}}(\sigma, f(\sigma), {\mathcal{P}^\star}^{\mathcal{O}}(\sigma)) = 1$, where $\sigma \leftarrow \mathcal{CRS}(1^n)$) with non-negligible probability, then there is an adversary $\mathcal{A}$ that finds (x, a) s.t. $\mathcal{O}(x, a) = g(x, a)$ with non-negligible probability. The adversary $\mathcal{A}$ works as follows. First, it runs $\sigma \leftarrow \mathcal{CRS}(1^n)$. Then it runs $(x, a, e, z) \leftarrow \mathcal{P}^\star(\sigma)$. Finally it outputs $(x, \mathcal{O}(x, a))$. From the contradicting assumption we know that $\mathcal{V}^{\mathcal{O}}(\sigma, f(\sigma), (a, e, z)) = 1$ with non-negligible probability. This implies that the transcript $(a, \mathcal{O}(x, a), z)$ is accepting with non-negligible probability. Since $x \notin L$ there exists only one e for which $(a, \mathcal{O}(x, a), z)$ is accepting. Therefore we have that with non-negligible probability it holds that $\mathcal{O}(x, a) = e$ (i.e., $\mathcal{O}(x, a) = g(x, a)$) and this contradicts the fact that any function g is evasive for an NPRO.

5 Our Transform: NIZK from HVZK

From the previous section we know that if we have a 3-round HVZK proof system with optimal soundness $\Pi_{L \vee \Lambda} = (\mathcal{P}_{L \vee \Lambda}, \mathcal{V}_{L \vee \Lambda})$ for polynomial relation

$$\hat{\mathcal{R}}_{L \vee \Lambda} = \{((x, \rho), w) : ((x, w) \in \mathcal{R}_L \wedge \rho \in \hat{\Lambda}) \vee ((\rho, \omega) \in \mathcal{R}_\Lambda \wedge x \in \hat{L})\}$$

that is also WI for polynomial relation

$$\mathcal{R}_{L \vee \Lambda} = \{((x, \rho), w) : ((x, w) \in \mathcal{R}_L \wedge \rho \in \Lambda) \vee ((\rho, \omega) \in \mathcal{R}_\Lambda \wedge x \in L)\}$$

we can apply the FS transform to make it non-interactive still preserving WI and soundness. To run the protocol a common hash function is needed and such a function is modeled as an NPRO in the proof of soundness.

Here we make use of the above result in order to transform a 3-round HVZK proof system with optimal soundness for an $\mathtt{NP}$-language L into a NIZK argument for L in the CRS model using an NPRO in the proof of soundness. The transformed NIZK argument $\Pi = (\mathcal{P}, \mathcal{V})$ is described below.

Common input: instance x for an $\mathtt{NP}$-language L.
Private input of $\mathcal{P}$: w s.t $(x, w) \in \mathcal{R}_L$.
Common reference string: $\mathcal{CRS}$ on input 1^n runs $\rho \leftarrow S_\Lambda(1, 1^n)$ where Λ is an membership-hard language and samples a key s for a hash function family H. Then it sets $\sigma = (\rho, s)$.

$\mathcal{P} \rightarrow \mathcal{V}$: $\mathcal{P}$ executes the following steps:

1. $a \leftarrow \mathcal{P}_{L\vee\Lambda}((x, \rho), w)$;
2. $e \leftarrow H_s(x, a)$;
3. $z \leftarrow \mathcal{P}_{L\vee\Lambda}((x, \rho), w, a, e)$;
4. send $\pi = (a, e, z)$ to $\mathcal{V}$.

$\mathcal{V}'s$ output: $\mathcal{V}$ accepts if and only if $\mathcal{V}_{L\vee\Lambda}((x, \rho), a, e, z) = 1$ and $e = H_s(x, a)$.

In our construction we suppose that the challenge length of Π_Λ is n, where n denotes the security parameter. Therefore to use the OR composition of [18] we need to consider a 3-round public-coin proof system with HVZK and optimal soundness Π_L for $\mathcal{R}_L$ that has challenge length n and therefore soundness error 2^{-n}). This is not a problem because we can use Theorem 2 to transform every 3-round public-coin proof system with HVZK and optimal soundness with challenge n' (where $n' \neq n$) to another one with challenge length n. More precisely, if $n' > n$ we can use Lemma 2 to reduce n' to n almost for free. If $n' < n$ we need to use Lemma 1, therefore we have to run multiple executions of Π_L to apply the OR composition of [18]. Notice that this potential computational effort is implicit also for the FS transform and for Lindell's transform. Indeed if the original 3-round public-coin proof system with HVZK and optimal soundness has just a one-bit (or in general a short) challenge then clearly the resulting NIZK is not sound. Therefore the parallel repetition of the 3-round public-coin proof system with HVZK and optimal soundness is required before applying the transform in order to reduce the soundness error (see Section 2.2).

Theorem 7. *Let $\Pi_{L\vee\Lambda}$ be a 3-round public-coin proof system for polynomial relation $\hat{\mathcal{R}}_{L\vee\Lambda}$ that is WI for polynomial relation $\mathcal{R}_{L\vee\Lambda}$. Then Π is zero knowledge for $\mathcal{R}_L$ in the CRS model.*

Proof. The simulator S works as follows:

1. S on input 1^n, runs $(\rho, \omega) \leftarrow S_\Lambda(0, 1^n)$; samples a key s for a hash function and sets $\sigma = \{\rho, s\}$ and outputs σ.
2. S on input σ, ω and x_i (for every $i = 1, \ldots, p(n)$) computes $a \leftarrow \mathcal{P}_{L\vee\Lambda}((x_i, \rho), \omega)$, $e \leftarrow H_s(x_i, a)$ and $z \leftarrow \mathcal{P}_{L\vee\Lambda}((x_i, \rho), \omega, a, e)$. It outputs $\pi_i = (a, e, z)$.

We show that the output of S is computationally indistinguishable from a real transcript given in output by $\mathcal{P}$ in a real execution of Π through the following hybrids games.

1. $\mathcal{H}_0$ is the experiment $R_f(\mathcal{P}^f(n, p))$ (Definition 7).

2. $\mathcal{H}_1$ differs from $\mathcal{H}_0$ in the way that ρ is generated. Indeed in $\mathcal{H}_1$ we have that σ is computed by running $S_\Lambda(0, 1^n)$. The second output ω of S_Λ is not used. Clearly $\mathcal{H}_0$ and $\mathcal{H}_1$ are indistinguishable otherwise the membership-hard property of Λ would be contradicted. More details on this reduction will be given below.
3. $\mathcal{H}_2$ differs from $\mathcal{H}_1$ just on the witness used by $\mathcal{P}_{L\vee\Lambda}$. Indeed now ω is used as witness. The WI property of $\Pi_{L\vee\Lambda}$ guarantees that $\mathcal{H}_2$ can not be distinguished from $\mathcal{H}_1$. More details on this reduction will be given below. Notice that $\mathcal{H}_2$ corresponds to the simulation.

$\mathcal{H}_0 \approx \mathcal{H}_1$: If there exists a malicious verifier $\mathcal{V}^\star$ that distinguishes between $\mathcal{H}_0$ and $\mathcal{H}_1$, then there exists an adversary $\mathcal{A}$ that breaks the membership-hard property of Λ. The reduction works as follows.

1. $\mathcal{A}$ queries the challenger of S_Λ that sends back ρ.
2. $\mathcal{A}$ samples a key s for a hash function family H and sets $\sigma = \{\rho, s\}$.
3. $\mathcal{A}$ on input $(x_i, w_i) \in \mathcal{R}_L$ for $i = 1, \dots, p(n)$ computes the following steps:
   3.1. compute $a_i \leftarrow \mathcal{P}_{L\vee\Lambda}((x_i, \rho), w_i)$;
   3.2. compute $e_i \leftarrow H_s(x_i, a_i)$;
   3.3. compute $z_i \leftarrow \mathcal{P}_{L\vee\Lambda}((x_i, \rho), w_i, a_i, e_i)$;
   3.4. set $\pi_i = (a_i, e_i, z_i)$;
   3.5. set $\boldsymbol{x} = x_1, \dots, x_i$ and $\boldsymbol{\pi} = \pi_1, \dots, \pi_i$.
4. $\mathcal{A}$ sends $\sigma, \boldsymbol{x}, \boldsymbol{\pi}$ to $\mathcal{V}^\star$.
5. $\mathcal{A}$ outputs the output of $\mathcal{V}^\star$.

We now observe that if the challenger of a sampling algorithm S_Λ sends $\rho \notin \Lambda$ we are in $\mathcal{H}_0$ otherwise we are in $\mathcal{H}_1$. This implies that $\mathcal{H}_0 \approx \mathcal{H}_1$.

$\mathcal{H}_1 \approx \mathcal{H}_2$: If there exists a distinguisher $\mathcal{V}^\star$ that distinguishes between $\mathcal{H}_1$ and $\mathcal{H}_2$, then there exists an adversary $\mathcal{A}$ against the adaptive NIWI property of $\Pi_{L\vee\Lambda}$, therefore contradicting Theorem 5. The reduction works as follows.

1. $\mathcal{A}$ runs $(\rho, \omega) \leftarrow S_\Lambda(0, 1^n)$, samples a key s for a hash function and sets $\sigma = \{\rho, s\}$.
2. $\mathcal{A}$ has on input a PPT function $f = (f_1, f_2)$ and defines $f' = (f'_1, f'_2)$ as follows:
   $f'(\sigma, \boldsymbol{t}, \boldsymbol{\pi})$ on input a CRS σ, a vector of theorems $\boldsymbol{t} = (x_1, \rho), \dots, (x_{p(n)}, \rho)$ and a vector of proofs $\boldsymbol{\pi} = \pi_1, \dots, \pi_{p(n)}$ returns $(f_1(\sigma, \boldsymbol{x}, \boldsymbol{\pi}), \rho), (f_2(\sigma, \boldsymbol{x}, \boldsymbol{\pi}), \omega)$.
3. $\mathcal{A}$ interacts with the challenger of adaptive NIWI, using f', in order to obtain x_i, $\pi_i = \{a_i, e_i, z_i\}$, for $i = 1, \dots, p(n)$.
4. $\mathcal{A}$ sets $\boldsymbol{x} = x_1, \dots, x_{p(n)}$ and $\boldsymbol{\pi} = \pi_1, \dots, \pi_{p(n)}$.
5. $\mathcal{A}$ sends $\sigma, \boldsymbol{x}, \boldsymbol{\pi}$ to $\mathcal{V}^\star$ and outputs the output of $\mathcal{V}^\star$.

We now observe that if the challenger of NIWI chooses the first witness w_i we are in $\mathcal{H}_1$ otherwise we are in $\mathcal{H}_2$. This implies that $\mathcal{H}_1 \approx \mathcal{H}_2$. We can thus conclude that $\mathcal{H}_0 \approx \mathcal{H}_1 \approx \mathcal{H}_2$ and therefore the output of S is computational indistinguishable from a real transcript.

Theorem 8. *Let $\Pi_{L \vee \Lambda}$ be a 3-round public-coin HVZK proof system with optimal soundness for relation $\mathcal{R}_{L \vee \Lambda}$, and WI for relation $\hat{\mathcal{R}}_{L \vee \Lambda}$, and let H be an NPRO. Then, Π is a non-interactive argument system with adaptive soundness for the relation $\mathcal{R}_L$ in the CRS model using the NPRO model for soundness.*

Proof. The completeness of Π follows from the completeness of $\Pi_{L \vee \Lambda}$. In order to prove adaptive soundness we notice that an adversarial prover proving a false statement $x \in L$ can be directly reduced to an adversarial prover proving a false statement for $\Pi_{L \vee \Lambda}$ in the NPRO model. This contradicts Theorem 6. Indeed the only subtlety that is worthy to note is that when the adversarial prover runs the protocol, we have that the statement "$\rho \in \Lambda$" stored in the CRS is false, therefore if also the instance "$x \in L$" proved by the prover is false then the OR composition of the two statements is also false.

6 Details on Some Σ-Protocols

First of all we need to briefly introduce two Σ-protocols, one to prove that a tuple is a DH tuple ($\Pi_{\mathcal{DH}}$ [41]), and the other one to prove that two graphs are isomorphic ($\Pi_{\mathcal{GH}}$ [34]). Our comparison assumes that the CRS is a DH tuple $((G_{\text{CRS}}, q_{\text{CRS}}, p_{\text{CRS}}, g_{\text{CRS}}), A_{\text{CRS}}, B_{\text{CRS}}, C_{\text{CRS}})$ with p_{CRS} and q_{CRS} primes such that $p_{\text{CRS}} = 2q_{\text{CRS}} + 1$ and $|p_{\text{CRS}}| = 1024$. We distinguish two cases. In the first one the prover wants to prove that a tuple $((G, q, p, g), A, B, C)$ is a DH tuple, and in the other one the prover tries to convince the verifier that two graphs G_0 and G_1 with n vertices each are isomorphic.

A Σ-protocol for Diffie-Hellman tuples. We consider the following polynomial-time relation $\mathcal{R}_{\mathcal{DH}} = \{(((G, q, g), A = g^r, B = h, C = h^r), r) : B^r = C\}$ over cyclic groups G_q of prime-order q. Typically, G is the subgroup of quadratic residues of $\mathbb{Z}_p$ for prime $p = 2q + 1$. We next briefly describe Σ-protocol $\Pi_{\mathcal{DH}} = (\mathcal{P}_{\mathcal{DH}}, \mathcal{V}_{\mathcal{DH}})$ for $\mathcal{R}_{\mathcal{DH}}$.

Common input: instance x and language DH.
Private input of $\mathcal{P}_{\mathcal{DH}}$: r.
The protocol $\Pi_{\mathcal{DH}}$:

1. $\mathcal{P}_{\mathcal{DH}}$ picks $t \in \mathbb{Z}_q$ at random, computes and sends $a = g^t$, $b = h^t$ to $\mathcal{V}_{\mathcal{DH}}$;
2. $\mathcal{V}_{\mathcal{DH}}$ chooses a random challenge $e \in \mathbb{Z}_q$ and sends it to $\mathcal{P}_{\mathcal{DH}}$;
3. $\mathcal{P}_{\mathcal{DH}}$ computes and sends $z = t + er$ to $\mathcal{V}_{\mathcal{DH}}$;
4. $\mathcal{V}_{\mathcal{DH}}$ checks $g^z = a \cdot A^e$ `AND` $h^z = b \cdot C^e$ accepts if and only if it is the case.

We show the special HVZK simulator Sim for $\Pi_{\mathcal{DH}}$. Sim, on input x and a challenge e of length $|q| - 1$ executes the following steps:

1. randomly chooses $z \in \mathbb{Z}_q$;
2. computes $a = g^z \cdot A^{-e}$;
3. computes $b = h^z \cdot C^{-e}$.

Graph isomorphism. We show a Σ-protocol $\Pi_{\mathcal{GH}} = (\mathcal{P}_{\mathcal{GH}}, \mathcal{V}_{\mathcal{GH}})$ to prove that two graphs are isomorphic. Given two graphs G_0 and G_1, prover $\mathcal{P}_{\mathcal{GH}}$ wants to convince verifier $\mathcal{V}_{\mathcal{GH}}$ that he knows a permutation ϕ such that $\phi(G_0) = G_1$.

Common input: theorem $x = (G_0, G_1)$.
Private input of $\mathcal{P}_{\mathcal{GH}}$: ϕ.
The protocol $\Pi_{\mathcal{GH}}$:

1. $\mathcal{P}_{\mathcal{GH}}$ randomly chooses a permutation ψ and a bit $b \in \{0,1\}$, computes and sends $P = \psi(G_b)$;
2. $\mathcal{V}_{\mathcal{GH}}$ chooses and sends a random bit $b' \in \{0,1\}$ $\mathcal{P}_{\mathcal{GH}}$;
3. $\mathcal{P}_{\mathcal{GH}}$ sends the permutation τ to $\mathcal{V}_{\mathcal{GH}}$, where

$$\tau = \begin{cases} \psi & if\ b = b' \\ \psi\phi^{-1} & if\ b = 0, b' = 1 \\ \psi\phi & if\ b = 1, b' = 0 \end{cases}$$

4. $\mathcal{V}_{\mathcal{GH}}$ accepts if and only if $P = \tau(G_{b'})$.

Computational effort: two cases. We show a summary of the comparison among our transform and Lindell's transform in Tables 2 and 3. The cost is measured by considering the computations in terms of number of exponentiations made by $\mathcal{P}$ and of $\mathcal{V}$. In our comparison we consider that a CRS contains a DH tuple $((G_{\text{CRS}}, q_{\text{CRS}}, p_{\text{CRS}}, g_{\text{CRS}}), A_{\text{CRS}}, B_{\text{CRS}}, C_{\text{CRS}})$ with $|p_{\text{CRS}}| = n = 1024$, with security parameter n (therefore $|q_{\text{CRS}}| = 1023$). We consider two cases. In the first one we use the NIZK argument to prove that a tuple $((G, q, p, g), A, B, C)$ is a DH tuple; in particular we take in account two sub-cases: when $p = 1024$ and when $p = 2048$. In the second case we use the NIZK argument to prove the isomorphism between two graphs G_0 and G_1, and we assume that $k = n^2$ bits are needed to represent a graph with n vertices. We stress that Lindell's transform needs to commit the first round of a Σ-protocol (plus the instance to be proved, but for our comparison we ignore that the instance has to be committed) associated to the language that we take into account (the language of the DH tuples or the language of the isomorphic graphs). Therefore, using the described CRS, to commit to a string of 1023 bit, 4 exponentiations are required. This is a consequence of the fact that the commitment is made by executing the simulator associated with $\Pi_{\mathcal{DH}}$ (with $|q_{\text{CRS}}| = 1023$).

Case 1: proving that a tuple is a DH tuple.

- [43]. When the instance to be proved is $((G, q, p, g), A, B, C)$ with $p = 1024$, the prover $\mathcal{P}$ needs to compute $a = g^t$, $b = h^t$ (as describe before) and needs to commit to them. The total size of a and b is 2048 bits, therefore to commit to 2048 bits we need to execute the DM commitment 3 times. This implies that the prover needs to compute $3 \cdot 4$ exponentiations mod p_{CRS} and 2 exponentiations mod p. The verifier $\mathcal{V}$ needs to checks if open of the

DM commitments was correct, and also needs to compute $g^z = a \cdot A^e p$ and $h^z = b \cdot C^e$. For this reason the verifier needs to compute $3 \cdot 4$ exponentiations mod p_{CRS} plus 4 exponentiations mod p. With the same arguments we can count the amount of exponentiations needed to prove that the instance is a DH tuple with $p = 2048$.

- Our transform. When $|p| = 1024$ (resp., $|p| = 2048$) the prover need to run the simulator Sim of $\Pi_{\mathcal{DH}}$ with the instance $((G_{\text{CRS}}, q_{\text{CRS}}, p_{\text{CRS}}, g_{\text{CRS}}), A_{\text{CRS}}, B_{\text{CRS}}, C_{\text{CRS}})$ (this costs 4 exponentiations), also we need to compute $a = g^t$, $b = h^t$. The total number of exponentiations is 6 (2 exponentiations mod p, and 4 exponentiations mod p_{CRS}). The verifier needs to perform two times the verifier's algorithm for $\Pi_{\mathcal{DH}}$, one with the instance $((G_{\text{CRS}}, q_{\text{CRS}}, p_{\text{CRS}}, g_{\text{CRS}}), A_{\text{CRS}}, B_{\text{CRS}}, C_{\text{CRS}})$, the other one with the instance $((G, q, p, g), A, B, C)$, for a total amount of 4 exponentiations mod p_{CRS}, and 4 exponentiations mod p.

Case 2: Graph isomorphism.

- [43]. We consider that the instance to be proved is composed by two graphs (G_0, G_1). Also we assume that to represent one graph with n vertices $k = n^2$ bits are necessary. In this case we remark that because the security parameter is $n = 1024$ we need to execute n times the protocol $\Pi_{\mathcal{GH}}$ described before. For the described assumptions we have that the first round of $\Pi_{\mathcal{GH}}$ is $P = \sigma(G_b)$ and $|P| = n^2$. Therefore the prover needs to run n executions of the DM commitment function to commit to P, where each of them costs 4 exponentiations. Also we need to execute n iteration of this process, for a total amount of $4n^2$ exponentiations mod p_{CRS}. Even in this case the verifier needs to checks if all opens with respect to the n commitments are correctly computed for a total amount of $4n^2$ exponentiations mod p_{CRS}.
- Our transform. In this case the prover $\mathcal{P}$ computes only 2 exponentiations mod p to compute the first round of $\Pi_{\mathcal{DH}}$. The verifier runs the verifier's algorithm of $\Pi_{\mathcal{DH}}$ that costs 4 exponentiations mod p.

7 Acknowledgments

We thank Alessandra Scafuro and Berry Schoenmakers for various useful discussions on Σ-protocols. An updated version of this work appears in [17].

References

1. Almeida, J.B., Bangerter, E., Barbosa, M., Krenn, S., Sadeghi, A., Schneider, T.: A certifying compiler for zero-knowledge proofs of knowledge based on sigma-protocols. In: ESORICS 2010. LNCS, vol. 6345, pp. 151–167. Springer (2010)
2. Barak, B., Canetti, R., Nielsen, J.B., Pass, R.: Universally composable protocols with relaxed set-up assumptions. In: 45th Symposium on Foundations of Computer Science (FOCS 2004), 17-19 October 2004, Rome, Italy (2004)

3. Bellare, M., Rogaway, P.: Random oracles are practical: A paradigm for designing efficient protocols. In: CCS '93, Proceedings of the 1st ACM Conference on Computer and Communications Security, Fairfax, Virginia, USA, November 3-5, 1993. pp. 62–73 (1993)
4. Bitansky, N., Dachman-Soled, D., Garg, S., Jain, A., Kalai, Y.T., López-Alt, A., Wichs, D.: Why "Fiat-Shamir for Proofs" Lacks a Proof, pp. 182 – 201. LNCS, Springer Berlin Heidelberg (2013/01/01/ 2013)
5. Blum, M., De Santis, A., Micali, S., Persiano, G.: Noninteractive zero-knowledge. SIAM J. Comput. 20(6), 1084–1118 (1991)
6. Blum, M., Feldman, P., Micali, S.: Non-interactive zero-knowledge and its applications. In: Proceedings of the 20th Annual ACM Symposium on Theory of Computing, May 2-4, 1988, Chicago, Illinois, USA. pp. 103–112 (1988)
7. Canetti, R., Goldreich, O., Halevi, S.: The random oracle methodology, revisited. In: Proceedings of the Thirtieth Annual ACM Symposium on the Theory of Computing, Dallas, Texas, USA, May 23-26, 1998. pp. 209–218 (1998)
8. Canetti, R., Goldreich, O., Halevi, S.: The random oracle methodology, revisited. J. ACM 51(4), 557–594 (Jul 2004)
9. Canetti, R., Lin, H., Paneth, O.: Public-coin concurrent zero-knowledge in the global hash model. In: TCC. pp. 80–99 (2013)
10. Catalano, D., Dodis, Y., Visconti, I.: Mercurial commitments: Minimal assumptions and efficient constructions. In: Theory of Cryptography, Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006, Proceedings. LNCS, vol. 3876, pp. 120–144. Springer (2006)
11. Catalano, D., Visconti, I.: Hybrid trapdoor commitments and their applications. In: Automata, Languages and Programming, 32nd International Colloquium, ICALP 2005, Lisbon, Portugal, July 11-15, 2005, Proceedings. pp. 298–310 (2005)
12. Catalano, D., Visconti, I.: Hybrid commitments and their applications to zero-knowledge proof systems. Theor. Comput. Sci. 374(1-3), 229–260 (2007)
13. Chaidos, P., Groth, J.: Making sigma-protocols non-interactive without random oracles. In: PKC 2015. pp. 650–670 (2015)
14. Chase, M., Lysyanskaya, A.: On signatures of knowledge. In: Dwork, C. (ed.) CRYPTO 2006, LNCS, vol. 4117, pp. 78–96. Springer Berlin Heidelberg (2006)
15. Ciampi, M., Persiano, G., Scafuro, A., Siniscalchi, L., Visconti, I.: Improved OR composition of Sigma-protocols. IACR Cryptology ePrint Archive 2015, 810 (2015), `http://eprint.iacr.org/2015/810`
16. Ciampi, M., Persiano, G., Scafuro, A., Siniscalchi, L., Visconti, I.: Improved OR composition of sigma-protocols. In: Theory of Cryptography - 13th Theory of Cryptography Conference, TCC 2016-A, Tel Aviv, Israel, January 10-13, (2016)
17. Ciampi, M., Persiano, G., Siniscalchi, L., Visconti, I.: A transform for NIZK almost as efficient and general as the Fiat-Shamir transform without programmable random oracles. IACR Cryptology ePrint Archive 770 (2015), `http://eprint.iacr.org/2015/770`
18. Cramer, R., Damgård, I., Schoenmakers, B.: Proofs of partial knowledge and simplified design of witness hiding protocols. In: Desmedt, Y. (ed.) CRYPTO '94, LNCS, vol. 839, pp. 174–187. Springer Berlin Heidelberg (1994)
19. Damgård, I.: Efficient concurrent zero-knowledge in the auxiliary string model. In: Advances in Cryptology - EUROCRYPT 2000, International Conference on the Theory and Application of Cryptographic Techniques, Bruges, Belgium, May 14-18, 2000, Proceeding. LNCS, vol. 1807, pp. 418–430. Springer (2000)
20. Damgård, I.: On Σ-protocol. `http://www.cs.au.dk/~ivan/Sigma.pdf` (2010)

21. Damgård, I., Fazio, N., Nicolosi, A.: Non-interactive zero-knowledge from homomorphic encryption. In: Theory of Cryptography, Third Theory of Cryptography Conference, TCC, 2006, New York, NY, USA, March 4-7, 2006 (2006)
22. Damgård, I., Groth, J.: Non-interactive and reusable non-malleable commitment schemes. In: Proceedings of the 35th Annual ACM Symposium on Theory of Computing, June 9-11, 2003, San Diego, CA, USA. pp. 426–437 (2003)
23. De Santis, A., Crescenzo, G.D., Ostrovsky, R., Persiano, G., Sahai, A.: Robust non-interactive zero knowledge. In: Advances in Cryptology - CRYPTO 2001, 21st Annual International Cryptology Conference, Santa Barbara, California, USA, August 19-23, 2001, Proceedings. pp. 566–598 (2001)
24. De Santis, A., Micali, S., Persiano, G.: Non-interactive zero-knowledge proof systems. In: Advances in Cryptology - CRYPTO '87, A Conference on the Theory and Applications of Cryptographic Techniques, Santa Barbara, California, USA, August 16-20, 1987, Proceedings. pp. 52–72 (1987)
25. Di Crescenzo, G., Visconti, I.: Concurrent zero knowledge in the public-key model. In: Automata, Languages and Programming, 32nd International Colloquium, ICALP 2005, Lisbon, Portugal, July 11-15, 2005. LNCS, vol. 3580, pp. 816–827. Springer (2005)
26. Dodis, Y.: G22.3220-001/g63.2180 Advanced Cryptography - Lecture 3 (Fall 2009)
27. Dwork, C., Naor, M.: Zaps and their applications. In: 41st Annual Symposium on Foundations of Computer Science, FOCS 2000, 12-14 November 2000, Redondo Beach, California, USA. pp. 283–293 (2000)
28. Dwork, C., Naor, M.: Zaps and their applications. SIAM J. Comput. 36(6), 1513–1543 (2007)
29. Feige, U., Lapidot, D., Shamir, A.: Multiple non-interactive zero knowledge proofs based on a single random string. In: 31st Annual Symposium on Foundations of Computer Science, St. Louis, Missouri, USA, October 22-24, 1990, Volume I. pp. 308–317 (1990)
30. Feige, U., Lapidot, D., Shamir, A.: Multiple non-interactive zero knowledge proofs under general assumptions. SIAM J. on Computing 29(1), 1–28 (1999)
31. Fiat, A., Shamir, A.: How to prove yourself: Practical solutions to identification and signature problems. In: CRYPTO '86, Santa Barbara, California, USA, 1986, Proceedings. pp. 186–194 (1986)
32. Garay, J.A., MacKenzie, P., Yang, K.: Strengthening zero-knowledge protocols using signatures. Journal of Cryptology 19(2), 169–209 (2006)
33. Garay, J.A., MacKenzie, P.D., Yang, K.: Strengthening zero-knowledge protocols using signatures. In: EUROCRYPT 2003. LNCS, vol. 2656, pp. 177–194. Springer (2003)
34. Goldreich, O., Micali, S., Wigderson, A.: Proofs that yield nothing but their validity and a methodology of cryptographic protocol design. In: 27th Annual Symposium on Foundations of Computer Science, Toronto, Canada, 27-29 October 1986. pp. 174–187 (1986)
35. Goldwasser, S., Kalai, Y.T.: On the (in)security of the fiat-shamir paradigm. In: 44th Symposium on Foundations of Computer Science (FOCS 2003), 11-14 October 2003, Cambridge, MA, USA, Proceedings. pp. 102–113 (2003)
36. Goldwasser, S., Micali, S., Rackoff, C.: The knowledge complexity of interactive proof-systems. In: Proceedings of the 17th Annual ACM Symposium on Theory of Computing, May 6-8, 1985, Providence, Rhode Island, USA. pp. 291–304 (1985)
37. Goldwasser, S., Micali, S., Rackoff, C.: The knowledge complexity of interactive proof systems. SIAM J. Comput. 18(1), 186–208 (1989)

38. Groth, J.: Honest verifier zero-knowledge arguments applied. Dissertation Series DS-04-3, BRICS. PhD thesis. xii+119 pp (2004)
39. Groth, J., Ostrovsky, R., Sahai, A.: Perfect non-interactive zero knowledge for NP. In: EUROCRYPT 2006. pp. 339–358 (2006)
40. Groth, J., Sahai, A.: Efficient non-interactive proof systems for bilinear groups. In: EUROCRYPT 2008. pp. 415–432 (2008)
41. Hazay, C., Lindell, Y.: Efficient Secure Two-Party Protocols - Techniques and Constructions. Information Security and Cryptography, Springer (2010)
42. Lindell, Y.: An efficient transform from Sigma Protocols to NIZK with a CRS and non-programmable random oracle. Cryptology ePrint Archive, Report 2014/710 (2014), `http://eprint.iacr.org/2014/710/20150906:203011`
43. Lindell, Y.: An efficient transform from Sigma protocols to NIZK with a CRS and non-programmable random oracle. In: Theory of Cryptography - 12th Theory of Cryptography Conference, TCC 2015, Warsaw, Poland, March 23-25, 2015, Proceedings, Part I. pp. 93–109 (2015)
44. Micciancio, D., Petrank, E.: Simulatable commitments and efficient concurrent zero-knowledge. In: EUROCRYPT 2003. pp. 140–159 (2003)
45. Naor, M., Yung, M.: Public-key cryptosystems provably secure against chosen ciphertext attacks. In: Proceedings of the 22nd Annual ACM Symposium on Theory of Computing, May 13-17, 1990, Baltimore, Maryland, USA. pp. 427–437 (1990)
46. Ostrovsky, R., Pandey, O., Visconti, I.: Efficiency preserving transformations for concurrent non-malleable zero knowledge. In: Theory of Cryptography, 7th Theory of Cryptography Conference, TCC 2010, Zurich, Switzerland, February 9-11, 2010. LNCS, vol. 5978, pp. 535–552. Springer (2010)
47. Pass, R.: On deniability in the common reference string and random oracle model. In: CRYPTO 2003 (2003)
48. Sahai, A.: Non-malleable non-interactive zero knowledge and adaptive chosen-ciphertext security. In: 40th Annual Symposium on Foundations of Computer Science, FOCS '99, 17-18 October, 1999, New York, NY, USA. pp. 543–553 (1999)
49. Scafuro, A., Visconti, I.: On round-optimal zero knowledge in the bare public-key model. In: EUROCRYPT 2012s. LNCS, vol. 7237, pp. 153–171. Springer (2012)
50. Schnorr, C.: Efficient identification and signatures for smart cards. In: Brassard, G. (ed.) CRYPTO '89, LNCS, vol. 435, pp. 239–252. Springer New York (1989)
51. Visconti, I.: Efficient zero knowledge on the internet. In: Automata, Languages and Programming, 33rd International Colloquium, ICALP 2006, Venice, Italy, July 10-14, 2006, Proceedings, Part II. pp. 22–33 (2006)
52. Wee, H.: Zero knowledge in the random oracle model, revisited. In: ASIACRYPT 2009. pp. 417–434 (2009)
53. Yung, M., Zhao, Y.: Interactive zero-knowledge with restricted random oracles. In: Theory of Cryptography, Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006, Proceedings. pp. 21–40 (2006)
54. Yung, M., Zhao, Y.: Generic and practical resettable zero-knowledge in the bare public-key model. In: EUROCRYPT 2007. LNCS, vol. 4515, pp. 129–147. Springer (2007)

A Dual Mode Commitments and the Need for *Strong* Σ-protocols

The following definition of a dual-mode commitment scheme (DMCS, in short) is from [43].

Definition 9 ([43]). *A* dual-mode commitment scheme (DMCS) *is a tuple of PPT algorithms* $(\mathsf{GenCRS}, \mathsf{Com}, \mathsf{Scom})$ *such that:*

- $\mathsf{GenCRS}(1^n)$ *outputs a common reference string, denoted by* ρ.
- $(\mathsf{GenCRS}, \mathsf{Com})$*: when* $\rho \leftarrow \mathsf{GenCRS}(1^n)$ *and* $m \in \{0,1\}^n$*, algorithm* $\mathsf{Com}_\rho(m; r)$ *with randomness* r *is a non-interactive perfectly-binding commitment scheme.*
- $(\mathsf{Com}, \mathsf{Scom})$*: For every PPT adversary* $\mathcal{A}$ *and every polynomial* $p(\cdot)$*, the output of the following two experiments is computationally indistinguishable:*

$Real_{\mathsf{Com},\mathcal{A}}(1^n)$	$Simulation_{\mathsf{Scom}}(1^n)$
– $\rho \leftarrow \mathsf{GenCRS}(1^n)$ – For $i = 1, \ldots, p(n)$: 1. $m_i \leftarrow \mathcal{A}(\rho, \boldsymbol{c}, \boldsymbol{r})$ 2. $r_i \leftarrow \{0,1\}^{\mathtt{poly}(n)}$ 3. $c_i = \mathsf{Com}_\rho(m_i; r_i)$ 4. Set $\boldsymbol{c} = c_1, \ldots, c_i$ and $\boldsymbol{r} = r_1, \ldots, r_i$ – Output $\mathcal{A}(\rho, m_1, r_1, \ldots, m_{p(n)}, r_{p(n)})$	– $\rho \leftarrow \mathsf{Scom}(1^n)$ – For $i = 1, \ldots, p(n)$: 1. $c_i \leftarrow \mathsf{Scom}$ 2. $m_i \leftarrow \mathcal{A}(\rho, \boldsymbol{c}, \boldsymbol{r})$ 3. $r_i \leftarrow \mathsf{Scom}(m_i)$ 4. Set $\boldsymbol{c} = c_1, \ldots, c_i$ and $\boldsymbol{r} = r_1, \ldots, r_i$ – Output $\mathcal{A}(\rho, m_1, r_1, \ldots, m_{p(n)}, r_{p(n)})$

Membership-hard languages with efficient sampling. Lindell defines a membership-hard language Λ as a language such that one can efficiently sample both instances that belong to the language and instances that do not belong to the language. Still distinguishing among these two types of instances is hard. This is formalized through a sampling algorithm S_Λ that on input a bit b outputs an instance $\rho \in \Lambda$ along with a witness ω when $b = 0$, and outputs an instance $\rho \notin \Lambda$ otherwise. No polynomial-time distinguisher on input ρ can guess b with probability non-negligibly better than $1/2$. Let S_Λ^ρ denote the instance part of the output (i.e., without the witness when b is 0).

Definition 10 ([43]). *Let* Λ *be a language. We say that* Λ *is membership-hard with efficient sampling if there exists a PPT sampler* S_Λ *such that for every PPT distinguisher* $\mathcal{D}$ *there exists a negligible function* μ *such that:*

$$|\mathrm{Prob}\,[\,\mathcal{D}(S_\Lambda^\rho(0, 1^n), 1^n) = 1\,] - \mathrm{Prob}\,[\,\mathcal{D}(S_\Lambda(1, 1^n), 1^n) = 1\,]\,| \leq \mu(n).$$

There are several popular membership-hard languages in literature. We will in particular consider the one considered by Lindell in [43]: the language DH of Diffie-Hellman triples.

Lindell's construction of a DMCS from Σ-protocols. Let us describe Lindell's construction of a DMCS from any membership-hard language Λ admitting a Σ-protocol $\Pi_\Lambda = (\mathcal{P}_\Lambda, \mathcal{V}_\Lambda)$ with simulator Sim_Λ for perfect special HVZK.

Regular ρ generation: Run sampler S_Λ for Λ with input $(1, 1^n)$ and receive back ρ (recall that $\rho \notin \Lambda$).

Commitment: To commit to a value $m \in \{0,1\}^n$ with randomness r, Com sets $e = m$, runs $\mathsf{Sim}_\Lambda(\rho, e)$ with randomness r and obtains (a, z). The output of Com is the commitment $c = a$ and the decommitment information (e, r).

Decommitment: To decommit, provide e, z and the receiver checks that $\mathcal{V}_\Lambda(\rho, a, e, z) = 1$.

Simulator Scom:

- On input 1^n, Scom runs the sampler S_Λ with input $(0, 1^n)$, and receives back (ρ, ω) (recall that $\rho \in \Lambda$ and ω is a witness to this fact). Then, Scom computes $a = \mathcal{P}_\Lambda(\rho, \omega)$, sets $c = a$ and outputs (c, ρ).
- On input $m \in \{0,1\}^n$, Scom sets $e = m$ and outputs $z = \mathcal{P}_\Lambda(\rho, \omega, a, e)$.

A.1 A Subtlety in Lindell's Construction: the Need of Strong Σ-protocols

We now discuss a subtlety in the construction of a DMCS from any Σ-protocol for a membership-hard language given in [43]. We stress that the content of this section does not apply when considering [42].

We observe that the construction of a DMCS from any Σ-protocol for a membership-hard language given in [43] works when the Σ-protocol is equipped with a simulator such that when the simulator gets as randomness the 3rd round of the prover, then the simulator is able to output the *same* first round of the prover. This special property has been investigated in [26] where it was called *strong* perfect special HVZK. In more details, a Σ-protocol is strong perfect special HVZK if it admits a simulator Sim that on input any challenge e outputs a transcript (a, e, z) that is perfectly indistinguishable from the distribution of the transcript generated by the prover when the challenge is e, but in addition it is required that the transcript is computed by sampling the 3rd round uniformly at random. The strong perfect special HVZK property is formalized below.

Definition 11 ([26]). *The special perfect HVZK property is strong if there exists a PPT simulator* Sim *for the special perfect HVZK property that on input* $x \in L_\mathcal{R}$ *and a challenge "*e*" works by sampling the 3rd round "*z*" uniformly at random and then computing the 1st round "*a*" deterministically from "*x, e*" and "*z*".*

Lindell's construction of a DMCS showed in [43] requires a simulator for strong perfect special HVZK.

A Σ-protocol Π_{DH} for DH. Now we show an artificial but useful example that shows a Σ-protocol with a simulator Sim for perfect special HVZK that however does not works if strong perfect special HVZK is desired.

The most widely used Σ-protocol $\Pi_{DH} = (\mathcal{P}_{DH}, \mathcal{V}_{DH})$ for the language DH consists in running in parallel two instances of a Σ-protocol for $DLog$ each proving knowledge a discrete logarithm. The two instances are linked together by having the verifier send the same challenge and expecting to receive the same third-round message. Schnorr's protocol [50] constitutes a natural choice for a Σ-protocol for $DLog$.

Consider instead instantiating the Σ-protocol for DH with the following Σ-protocol $\Pi_{DLog} = (\mathcal{P}_{DLog}, \mathcal{V}_{DLog})$ for proving knowledge of the discrete logarithm w of x with base g. $\mathcal{P}_{DLog}$ first selects another random group element x' along with its discrete logarithm w' to the base g and then sends x' to $\mathcal{V}_{DLog}$. Then $\mathcal{P}_{DLog}$ and $\mathcal{V}_{DLog}$ run two instances of Schnorr's Σ-protocol using the same challenge so that $\mathcal{P}_{DLog}$ proves to $\mathcal{V}_{DLog}$ knowledge of both w and w'. Clearly, Π_{DLog} is a Σ-protocol for $DLog$ (this comes from the fact that the AND of two Σ-protocols is still a Σ-protocol and from the fact that knowledge of a pair (w, w') implies knowledge of w) and, consequently, Π_{DH} instantiated with Π_{DLog} is a Σ-protocol for DH. Moreover notice that Π_{DLog} admits a simulator $\mathsf{Sim}^{\star}_{DLog}$ for perfect HVZK that uses the simulator of Schnorr's protocol to compute the transcript of the first instance, while it uses the prover of Schnorr's protocol for producing the transcript associated to x', after having selected x' along with a witness w' when the protocol starts. We now provide a formal description of this Σ-protocol.

More precisely we show a Σ-protocol $\Pi_{DLog} = (\mathcal{P}_{DLog}, \mathcal{V}_{DLog})$ for relation $\mathcal{R}_{DLog} = \{((\mathcal{G}, g, q, x), w) : x = g^w\}$ that is special perfect HVZK and such that there exists a simulator for special perfect HVZK that does not satisfy the requirement of *strong* perfect special HVZK of Π_{DLog} (see Def. 11).

Common Input: $(\mathcal{G}, g, q, x)$ and relation $\mathcal{R}_{DLog}$.
Input of $\mathcal{P}_{DLog}$: w t.c $((\mathcal{G}, g, q, x), w) \in \mathcal{R}_{DLog}$.
The protocol Π_{DLog}:

1. $\mathcal{P}_{DLog}$ chooses r_0, r_1, w_1 at random from $\mathcal{Z}_q$, and g_1 at random from $\mathcal{G}$. Then it computes $(a_0, a_1) = (g^{r_0}, g_1^{r_1})$, and $x_1 = g_1^{w_1}$. $\mathcal{P}_{DLog}$ sends (a_0, g_1, x_1, a_1) to $\mathcal{V}_{DLog}$.
2. $\mathcal{V}_{DLog}$ chooses a random challenge $e \leftarrow \{0,1\}^l$ (where $2^l < q$) and sends e to $\mathcal{P}_{DLog}$.
3. $\mathcal{P}_{DLog}$ computes $z_0 = r_0 + ew$ and $z_1 = r_1 + ew_1$ it sends (z_0, z_1) to $\mathcal{V}_{DLog}$.
4. $\mathcal{V}_{DLog}$ checks $g^{z_0} = a_0 x^e$ and $g_1^{z_1} = a_1 x_1^e$ accepts if and only if it is the case.

Special HVZK The simulator Sim of Π_{DLog} on input the theorem $(\mathcal{G}, g, q, x)$ and challenge e works as follows:

1. pick z_0, r_1, w_1 at random from $\mathcal{Z}_q$ and g_1 at random from $\mathcal{G}$.
2. compute $a_0 = g^{z_0} x^{-e}$ and $a_1 = g_1^{r_1}$.
3. compute $x_1 = g_1^{w_1}$ and $z_1 = r_1 + ew_1$.
4. return $(a_0, g_1, x_1, a_1, z_0, z_1)$.

Completeness. In order to see that completeness holds, observe that when $\mathcal{P}_{DLog}$ runs the protocol honestly we have:

$$g^{z_0} = g^{r_0+we} = g^{r_0} \cdot g^{we} = a_0 \cdot x^e \quad \text{and} \quad g_1^{z_1} = g_1^{r_1+w_1e} = g_1^{r_1} \cdot g_1^{w_1e} = a_1 \cdot x_1^e.$$

Special soundness. Let $(a_0, g_1, x_1, a_1, e, z_0, z_1)$ $(a_0, g_1, x_1, a_1, e', z_0', z_1')$ be a collision. We have that $g^{z_0} = a_0x^e$ and $g^{z_0'} = a_0x^{e'}$, and thus we have $g^{z_0-z_0'} = x^{e-e'}$ that implies that $x = g^{\frac{z_0-z_0'}{e-e'}}$, therefore $w = \frac{z_0-z_0'}{e-e'}$.

Special perfect HVZK. We now check that the transcript returned by Sim, on input the theorem $(\mathcal{G}, g, q, x)$ and challenge e, is identically distributed w.r.t. the transcript obtained from the interaction between $\mathcal{P}_{DLog}$ and $\mathcal{V}_{DLog}$, when the challenge is e. The transcript differs only in the computation of a_0 and z_0. In the case of the $\mathcal{P}_{DLog}$ $a_0 = g^{r_0}$ where r_0 is chosen uniformly at random and $z_0 = r_0 + ew$. Instead, Sim chooses z_0 uniformly at random and $r_0 = z_0 - ew$, therefore clearly Sim and $\mathcal{P}_{DLog}$ produce a_0 and z_0 with the same distribution.

Π_{DH} does not produce a DMCS. We observe that Lindell's construction of a DMCS from any Σ-protocol for a membership-hard language [43] does not seem to work when Π_{DH} is used as Σ-protocol. Indeed consider the steps of experiments $Real_{\mathsf{Com},\mathcal{A}}(1^n)$ and $Simulation_{\mathsf{Scom}}(1^n)$ in which $\mathcal{A}$ obtains as input $(\rho, \boldsymbol{c}, \boldsymbol{r})$ and consider iteration with $i = 2$ of the loop.

In $Real_{\mathsf{Com},\mathcal{A}}(1^n)$, $\mathcal{A}$'s view includes (m_1, r_1, c_1) and thus $\mathcal{A}$ can check that indeed c_1 is the output of $\mathsf{Com}(m_1; r_1)$. This means that in the above construction, c_1 is the first component of the pair given in output by $\mathsf{Sim}_\Lambda(\rho, e)$ when running with randomness r_1, and this is precisely the way in which c_1 was produced in Step 3 when $i = 1$. Therefore the check of $\mathcal{A}$ succeeds in $Real_{\mathsf{Com},\mathcal{A}}(1^n)$.

In $Simulation_{\mathsf{Scom}}(1^n)$, $\mathcal{A}$'s view includes (m_1, r_1, c_1) and thus $\mathcal{A}$ can still perform the check that c_1 is the output of $\mathsf{Com}(m_1; r_1)$ by running $\mathsf{Sim}_\Lambda(\rho, e)$ with randomness r_1. However, in this case it is *not* true that c_1 is computed by running $\mathsf{Com}(m_1; r_1)$. Indeed, in the execution of $Simulation_{\mathsf{Scom}}(1^n)$, c_1 is computed by running $c_1 \leftarrow \mathsf{Scom}$ and then r_1 is computed by running $r_1 \leftarrow \mathsf{Scom}(m_1)$. In the above construction Scom computes c_1 and r_1 as the 1st and 3rd messages that are computed by $\mathcal{P}_\Lambda$ when the challenge is m_1. Therefore whenever the 3rd round r_1 computed by $\mathcal{P}_\Lambda$ does not correspond to a randomness that can be given as input to $\mathsf{Sim}_\Lambda(\rho, m_1)$ to get the same c_1 computed by $\mathcal{P}_\Lambda$, we have that the check of $\mathcal{A}$ fails.

By noticing that the 3rd round r_1 of $\mathcal{P}_{DH}$ in Π_{DH} does not give any information about the random instance x' of $DLog$ that $\mathcal{P}'_{DH}$ would compute and that would be part of c_1, we have that there exists a simulator for DH, using internally $\mathsf{Sim}^\star_{DLog}$, that on input (ρ, m_1) and running with randomness r_1 computes c_1 only with negligible probability and thus the above $\mathcal{A}$ is a successful distinguisher of experiments $Real_{\mathsf{Com},\mathcal{A}}(1^n)$ and $Simulation_{\mathsf{Scom}}(1^n)$.

B An Optimal-Sound (and Not Special Sound) 3-Round Perfect Special HVZK Proof

In this section we show a 3-round public-coin perfect special HVZK proof system that is optimal sound and not special sound. First of all we briefly describe the Σ-protocol of [44] to prove that, given a commitment $\mathtt{com}$ and a message m, m is committed in $\mathtt{com}$. Then we show the protocol of [51] that is a modification of [44] and given a commitment $\mathtt{com}$ and a value Ψ, allows to prove that the discrete logarithm of Ψ is committed in $\mathtt{com}$.

In order to describe the protocol of [44] and [51] we consider two prime p and q s.t. $p = 2q + 1$, a group of order $\mathcal{G}$ of order q such that the DDH assumption is hard. Also we consider two random elements, g and h, taken from $\mathcal{G}$. We next describe Σ-protocol $\Pi_{Com} = (\mathcal{P}_{Com}, \mathcal{V}_{Com})$ of [44] for relation

$$\mathcal{R}_{Com} = \left\{ \left(\left((\mathcal{G}, q, g, h), v, \mathtt{com} = (\hat{g}, \hat{h}) \right), w \right) : \hat{g} = g^w, \hat{h} = h^{w+v} \right\}.$$

Common Input: $(\mathcal{G}, g, v, h, \mathtt{com} = (\hat{g}, \hat{h}), q)$ and relation $\mathcal{R}_{Com}$.
Input of $\mathcal{P}_{Com}$: w s.t. $((\mathcal{G}, v, g, h, \mathtt{com} = (\hat{g}, \hat{h}), q), w) \in \mathcal{R}_{Com}$.
The protocol Π_{Com}:

1. The prover $\mathcal{P}_{Com}$ chooses r from $\mathcal{Z}_q$ and sends $(\widetilde{g} = g^r, \widetilde{h} = h^r)$ to $\mathcal{V}_{Com}$;
2. The verifier $\mathcal{V}_{Com}$ chooses a random challenge $e \leftarrow \mathcal{Z}_q$ and sends e to $\mathcal{P}_{Com}$;
3. $\mathcal{P}_{Com}$ sends $z = ew + r$ to $\mathcal{V}_{Com}$;
4. $\mathcal{V}_{Com}$ checks that $\hat{g}^e \widetilde{g} = g^z$ and $\left(\frac{\hat{h}}{h^v}\right)^e \widetilde{h} = h^z$ accepts if and only if the checks are successful.

In [51] a similar protocol was used to prove that $\mathtt{com}$ is a commitment of the discrete logarithm of a value $\Psi \in \mathcal{G}$ with $h^\psi = \Psi$. Formally the protocol is for the $\mathtt{NP}$ language

$$L = \left\{ \left(\Psi = h^\psi, \mathtt{com} = (\hat{g} = g^w, \hat{h} = h^{w+\psi}) \right) : g, h \leftarrow \mathcal{G}, \psi \in \mathbb{Z}_q, w \in \mathbb{Z}_q \right\}$$

and for the corresponding relation

$$\mathcal{R}_L = \left\{ \left((\Psi = h^\psi, \mathtt{com} = (\hat{g} = g^w, \hat{h} = h^{w+\psi})), (w, \psi) \right) : g, h \leftarrow \mathcal{G}, \psi \in \mathbb{Z}_q, w \in \mathbb{Z}_q \right\}$$

The protocol follows Π_{Com} with the differences that the common input is $(\mathcal{G}, q, g, \Psi = h^\psi, h, \mathtt{com} = (\hat{g}, \hat{h})$ and that the verifier decide whether to accept or not checking if it holds that $\hat{g}^e \widetilde{g} = g^z$ and $\left(\frac{\hat{h}}{\Psi}\right)^e \widetilde{h} = h^z$. While this protocol preserves the perfect special HVZK property, it is not a proof of knowledge for $\mathcal{R}_L$ and neither special sound even though it still enjoys optimal soundness. We now proceed more formally.

Optimal soundness. We now consider an instance that is not in the NP language L, and show that, once the first round of the protocol is fixed, there exists only one challenge e s.t. the prover can answer successfully computing the third round z of the protocol. Consider the instance $\left(\Psi = h^{\psi}, \mathtt{com} = (\hat{g} = g^{w}, \hat{h} = h^{w+\psi\prime})\right) \notin L$ (with $\psi \neq \psi\prime$). Assume by contradiction that given the fist round of the protocol $(\widetilde{g}, \widetilde{h})$ there exist two distinct challenges e_0 and e_1 for which the prover can make the verifier accept with answers z_0, z_1 respectively. In the end we prove that $\psi = \psi\prime$.

Proof. Since the verifier accepts, it must be that for all $i \in \{0, 1\}$, the following checks are successful: $\hat{g}^{e_i}\widetilde{g} = g^{z_i}$ and $\left(\frac{\hat{h}}{\Psi}\right)^{e_i}\widetilde{h} = h^{z_i}$. It follows that $\hat{g}^{e_0-e_1} = g^{z_0-z_1}$ and $\left(\frac{\hat{h}}{\Psi}\right)^{e_0-e_1} = h^{z_0-z_1}$. Suppose that $h = g^{\omega}$, we get

$$\begin{aligned} g^{w\omega(e_0-e_1)} = \hat{g}^{(e_0-e_1)\omega} = g^{(z_0-z_1)\omega} = h^{(z_0-z_1)} &= (\hat{h}\cdot\Psi^{-1})^{e_0-e_1} \\ = h^{z_0-z_1} &= g^{\omega(w+\psi\prime-\psi)(e_0-e_1)}. \end{aligned}$$

Therefore, if $e_0 \neq e_1$ we get the contradiction that $\psi = \psi\prime$.

The protocol is not special sound for $\mathcal{R}_L$. To argue that the protocol of [51] is not special sound, we note that in order to compute a commitment com of the discrete logarithm of Ψ, knowledge of this discrete logarithm is not necessary since it is possible to compute $\mathtt{com} = (\hat{g}, h^{w}\cdot\Psi)$ with $w \in \mathbb{Z}_q$. Indeed, notice that the discrete logarithm ψ of Ψ is never used in the proof. Formally, we suppose that the protocol is special sound for the polynomial relation $\mathcal{R}_L$ and then construct an adversary $\mathcal{A}$ that, given $Y = g^{y} \in \mathcal{G}$, returns the discrete logarithm y of Y.

We have shown that there exist 3-round public-coin proof systems that are optimal sound and not special sound. It also easy to observe that special soundness implies optimal soundness. Indeed, consider an NP-Language L and a corresponding relation $\mathcal{R}_L$. All Σ-protocols for $\mathcal{R}_L$ must also be 3-round HVZK proofs for L with optimal soundness. If not, than the violation of optimal soundness ($\mathcal{P}^{\star}$ for a false statement can generate (a, c, z) and (a, c', z') with c' different from c and both accepting) implies directly also a violation of special soundness.