# On the Security of Classic Protocols for Unique Witness Relations

Yi Deng[1,2], Xuyang Song[1,2], Jingyue Yu[1,2], and Yu Chen[1,2](✉)

[1] State Key Laboratory of Information Security,
Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
[2] School of Cyber Security, University of Chinese Academy of Sciences,
Beijing, China

**Abstract.** We revisit the problem of whether the known classic constant-round public-coin argument/proof systems are witness hiding for languages/distributions with unique witnesses. Though strong black-box *impossibility* results are known, we provide some less unexpected *positive* results on the witness hiding security of these classic protocols:

- We give sufficient conditions on a hard distribution over *unique* witness NP relation for which all witness indistinguishable protocols (including all public-coin ones, such as ZAPs, Blum protocol and GMW protocol) are indeed witness hiding. We also show a wide range of cryptographic problems with unique witnesses satisfy these conditions, and thus admit constant-round public-coin witness hiding proof system.
- For the classic Schnorr protocol (for which the distribution of statements being proven seems not to satisfy the above sufficient conditions), we develop an embedding technique and extend the result of Bellare and Palacio to base the witness hiding property of the Schnorr protocol in the standalone setting on a *relaxed* version of one-more like discrete logarithm (DL) assumption, which essentially assumes there does not exist *instance compression* scheme for the DL problem, and show that breaking this assumption would lead to some surprising consequences, such as zero knowledge protocols for the AND-DL language with extremely efficient communication and highly non-trivial hash combiner for hash functions based on the DL problem. Similar results hold for the Guillou-Quisquater protocol.

## 1 Introduction

Witness hiding proof system, introduced by Feige and Shamir [12], is a relaxed yet natural notion of zero knowledge proof [15]. Instead of requiring an efficient simulation for the view of the verifier as in zero knowledge proof, witness hiding property only requires that, roughly speaking, the interaction with honest prover does not help the verifier compute any new witness for the statement being proven that he did not know before. One immediate application of such a security notion is identification: Witness hiding proof allows a prover to prove his identity

M. Abdalla and R. Dahab (Eds.): PKC 2018, LNCS 10770, pp. 589–615, 2018.
https://doi.org/10.1007/978-3-319-76581-5_20

without leaking the associated secret key, and this security notion is sufficient for preventing impersonation attack from malicious verifiers.

The witness hiding property of some practical protocols, which are usually not zero knowledge, is often proved via another beautiful and widely applicable notion of witness indistinguishability introduced in the same paper of [12]. A witness indistinguishable proof guarantees that if the statement has two independent witnesses, then the malicious verifier cannot tell which witness is being used by the prover in an execution of the protocol. The idea underlying the security proof of witness hiding via witness indistinguishability is as follows. Suppose that for a hard language, each instance has two witnesses and it is infeasible for an efficient algorithm, given one witness as input, to compute the other one, then the witness indistinguishable protocol is actually witness hiding with respect to such instances. This is because we can take one witness as input to play the role of honest prover and then use the verifier's ability of breaking witness hiding to either break witness indistinguishability of this protocol or obtain a new witness. Therefore, the parallelized version of 3-round public-coin classic protocols of [3,14] are witness hiding with respect to such languages.

What happens if the hard language consists of instances that have exactly one witness? This problem has turned out to be quite subtle. The Guillou-Quisquater [17] and the Schnorr [28] identification protocols are perhaps the best-known efficient protocols for unique witness relations, but their security has long remained open. On the positive side, Shoup [29] presented positive result that the Schnorr identification protocol is secure in the generic group model, and Bellare and Palacio [2] showed that the security of the Guillou-Quisquater and Schnorr identification protocols can be based on the so-called one-more RSA and one-more discrete logarithm assumptions, respectively [1,2]. These security proofs of course imply that the Schnorr and the Guillou-Quisquater identification protocols are witness hiding in the standalone setting where there is only a single execution of the protocol. However, the underlying assumptions/models are quite strong and non-standard.

Indeed, there is an obstacle in the way of basing constant-round public-coin protocols for unique witness relations on standard assumptions. As mentioned before, the basic approach to prove witness hiding of a protocol is to find an efficient way to exploit the power of the malicious verifier to break some hardness assumptions. For the instance that has exactly one witness, however, to exploit the power of the malicious verifier requires the reduction itself to know the unique witness to the statement being proven in the first place (by the soundness property of the protocol), which usually does not lead to a desired contradiction even if the malicious verifier does have the ability to break witness hiding of the protocol.

Haitner et al. [18] gave the first proof that constant-round public-coin witness hiding protocols for unique witness relations cannot be based on standard assumptions via some restricted types of black-box reductions. Pass [24] showed that if we further require witness hiding to hold under sequential repetition, then we can significantly strengthen the impossibility result of [18]. Some similar impossibility results on the problem whether we can base the aforementioned one-more discrete logarithm assumption on standard hardness assumption were also given in [24,30]. We would like to point out that these impossibility results may have some impact on other important problems. For example, in [23] Pass showed a deep connection between the problem of whether the classic constant-round public-coin proofs are witness hiding for all NP languages and the longstanding problem whether we can base one-way functions on NP-complete problem.

### 1.1 Our Contribution

Our main contribution reflects an optimistic point of view on the witness hiding security of the classic public-coin proof systems.

We observe that all previously known impossibility results [18,24] on the witness hiding of public-coin protocols make an *implicit* restriction (which has not been mentioned explicitly in the statements of their main results) on the black-box reduction: For a distribution $(\mathcal{X}, \mathcal{W})$ on an *unique* witness relation, for the proof of lower bound to go through, the (black-box) reduction $R$ is restricted to invoke the adversarial verifier $V^*$ *only* on instances in $\mathcal{X}$.[1]

This leaves a problem of whether one can get around these impossibility results by removing the above restriction on the black-box reduction. We provide a positive answer to this problem. Specifically, we develop an input-distribution-switching technique and prove that, for any hard language $L$, if a distribution $(\mathcal{X}, \mathcal{W})$ on a *unique* witness relation $R_L$ has an indistinguishable counterpart distribution over some *multiple* witnesses relation, then any witness indistinguishable protocols (including ZAPs and all known 3-round public-coin protocols, such as Blum protocol and GMW protocol) are indeed witness hiding for the unique witness distribution $(\mathcal{X}, \mathcal{W})$. We also show a wide range of cryptographic problems with unique witnesses satisfy the "if condition" of this result, and thus admit constant-round public-coin witness hiding proof system. This is the *first* positive result on the witness-hiding property of the classic protocols for unique witness relations.

For the classic Schnorr protocol (for which the distribution of statements being proven seems not to satisfy the above sufficient conditions), we develop an embedding technique and extend the result of [2] to base the witness hiding property of the standalone Schnorr (and Guillou-Quisquater) protocol based on a relaxed version of one-more like DL (RSA, respectively) assumption. To see the plausibility of our still-non-standard assumption, we follow the framework of [19] and introduce the notion of *tailored instance compression*, which captures the

[1] This restriction can be seen from the last paragraph "on the role of unique witness", page 7 of the full version (see http://www.cs.cornell.edu/~rafael/papers/schnorr.pdf) of [24]: "...(in the reduction) If the statement $x$ has a unique witness $w$, we can ensure that the extracted witness will be identical to the witness that the oracle A (which is $V^*$ in our setting) would have returned..".

essence of the known one-more like assumptions, and more importantly, provides new insight into the hardness of one-more DL/RSA problems and allows us to reveal some surprising consequences of breaking our version of the one-more like assumptions, including zero knowledge proofs with extremely low communication complexity for the AND-DL and AND-RSA languages and non-trivial hash combiner for hash functions based on DL problem.

We summarize our results in the Table 1.

**Table 1.** Our results for languages with unique witnesses compared to previous work. Here we refer to the impossibility results of further basing instance incompressibility/one-more assumptions on standard hard problems as "BB negative results/evidences", and refer to the surprising consequences of breaking these assumptions as "positive results/evidences" in favor of these assumptions. As we observe, the impossibility results of [18, 24] make an *implicit* restriction on the black-box reduction.

| | Security of Schnorr/GQ | Instance incompressibility/one-more assumptions | WH of PC protocols for unique witness R |
|---|---|---|---|
| BB negative results/evidences | [24] | [24, 30] | [18, 24] |
| Positive results/ evidences | [2]<br>This work (with relaxed assum.) | This work | This work |

## 1.2 Techniques

Input-distribution-switching technique: jumping out of the box. As mentioned before, the previously known impossibility results hold only with respect to *restricted* reduction. We introduce an *input-distribution-switching* technique to get around these impossibility results for general unique witness NP relations.

Suppose that, for a hard language $L_1$ with *unique* witness relation $R_{L_1}$, and a distribution ensemble $(\mathcal{X}^1, \mathcal{W}^1)$ over $R_{L_1}$, there exists a coupled distribution ensemble $(\mathcal{X}^2, \mathcal{W}^2)$ over relation $R_{L_2}$ of a language $L_2$ with *two or more* witnesses that is indistinguishable from $(\mathcal{X}^1, \mathcal{W}^1)$. What can we say about the security of the classic public-coin protocols for $(\mathcal{X}^1, \mathcal{W}^1)$? At least we know that such protocols are witness indistinguishable for $(\mathcal{X}^2, \mathcal{W}^2)$.

A very vague intuition behind this positive result is that, for the same malicious verifier $V^*$, if we invoke $V^*$ on both instances in $\mathcal{X}^1$ and $\mathcal{X}^2$, it should have the same behavior in these two settings since these instances are indistinguishable. This vague idea leads us to introduce the *input-distribution-switching technique*, which enables us to prove that if the ensembles $(\mathcal{X}^1, \mathcal{W}^1)$ and $(\mathcal{X}^2, \mathcal{W}^2)$ further satisfy the following properties:

- Given a sample $x$ from $\mathcal{X}^1$, it is hard to find the unique witness for $x$;
- For every $x$ in the support of $\mathcal{X}^2$, witnesses in $R_{L_2}(x)$ are uniformly distributed.

Then the classic constant-round public-coin protocols are actually witness hiding for $(\mathcal{X}^1, \mathcal{W}^1)$.

The proof of this result is a reduction of witness hiding for $(\mathcal{X}^1, \mathcal{W}^1)$ to witness indistinguishability for $(\mathcal{X}^2, \mathcal{W}^2)$, which is more complicated than one might imagine. See Sect. 3 for the detailed proof.

The idea of considering different types of distributions $\mathcal{X}^1$ and $\mathcal{X}^2$ on the common input already appeared in Goldreich's definition of strong witness indistinguishability [13], but there they do not require indistinguishability of $(\mathcal{X}^1, \mathcal{W}^1)$ and $(\mathcal{X}^2, \mathcal{W}^2)$ since such requirement on the witness distributions $\mathcal{W}^1$ and $\mathcal{W}^2$ would trivialize the definition of witness indistinguishability.

In our setting, the indistinguishability requirement on witness distributions $\mathcal{W}^1$ and $\mathcal{W}^2$ is helpful in achieving significant positive results on witness hiding protocols that bypass some previously known limitations. We give several examples of such distribution ensembles $(\mathcal{X}^1, \mathcal{W}^1)$ based on standard assumptions such as DDH, the existence of lossy trapdoor functions [25] and mixed commitments [9,16], and applying the above result we show the classic protocols of [3,11,14,16] are actually witness hiding under sequential repetition for a wide range of useful cryptographic problems with unique witnesses.

Embedding technique and the instance compression problem. Before proceeding to our embedding reduction, we recall the Schnorr protocol and Bellare and Palacio's security proof for it [2]. Let $\mathbb{G}$ be a group of prime order $q$ generated by $g$, the prover $P$ wants to convince the verifier $V$ of knowledge of the discrete logarithm (unique witness) $w \in \mathbb{Z}_q$ of an element $y = g^w \in \mathbb{G}$. To do so, $P$ first sends a random element $a = g^r \in \mathbb{G}$ to $V$, and upon receiving the $V$'s challenge $c \in \mathbb{Z}_q$, it answers with a value $z \in \mathbb{Z}_q$. $V$ accepts the proof if and only if $g^z = a \cdot y^c$. Note that, if $V$ finally outputs the witness $w \in \mathbb{Z}_q$ at the end of interaction, then we can build an algorithm $R$ solving two random discrete logarithm instances $y$ and $a$ at the same time if $R$ is allowed to make one query to the discrete logarithm solver oracle $\mathcal{O}_{\text{dlog}}$: $R$ have $y$ serve as the common input and $a$ as the first prover message, after receiving $V$'s challenge $c$, $R$ queries $\mathcal{O}_{\text{dlog}}$ on $a \cdot y^c$ and forwards the response $z$ from the oracle to the verifier; when $V$ outputs $w$, $R$ can solve the linear equation $z = r + cw \bmod q$ and obtain $r$. This useful observation was also exploited by Bellare and Palacio [2] to prove the security of the Schnorr protocol as an identification scheme under the hardness of one-more discrete logarithm problem.

We now show how to conduct embedding reduction $R$ that leads to better security proof based on a relaxed version of the one-more DL assumption.

Suppose we are given a set of discrete logarithm instances $(y_1, y_2, \ldots, y_\ell)$ to solve. For simplicity, we assume $\ell = 2^l$ for some integer $l$. The first part of $R$ is a *compressing* process. $R$ partitions them into $\ell/2$ pairs, for each pair of instances, one serving as the common input and the other serving as the first prover message in a session, and invokes $\ell/2$ incarnations of the verifier in parallel. After collecting $\ell/2$ challenges from the $\ell/2$ invocations of the verifier, $R$ has to solve $\ell/2$ new instances in order to answer each verifier. At this point, rather than querying $\mathcal{O}_{\text{dlog}}$ on these new instances, $R$ pauses all these interactions and partitions the new $\ell/2$ instances into $\ell/4$ pairs, and then repeats the above step and invokes $\ell/4$ incarnations of the verifier in parallel, and will get $\ell/8$ new instances to solve. Continuing to repeat this, by viewing each partial interaction with a verifier as a node we get a tree in which each node takes in two instances and outputs one instance. Finally, $R$ reaches the root and has only one instance to solve.

The second part of $R$ is an *unfolding* process. $R$ queries $\mathcal{O}_{\text{dlog}}$ on the root instance, then by using the verifier's power of breaking witness hiding as above, $R$ is able to solve the two instances flowing into this node. Note that, the two instances $R$ just solved will help it solve the four instances that flow into the two nodes at the level above the root (without making queries to oracle anymore), and repeating this process $R$ will solve all these $\ell$ instances $(y_1, y_2, \ldots, y_\ell)$. Observe that in the entire embedding reduction, $R$ makes only a single query (at the root of the tree) to $\mathcal{O}_{\text{dlog}}$ and solves all $\ell$ DL instances. This process is exemplified in Fig. 2.

The actual embedding reduction needs to make each invocation of the verifier independent by using the random self-reducibility of the discrete logarithm problem. As we will see, the quantity $\ell$ can be an arbitrarily large integer, or any polynomial when the verifier's success probability is close to 1. Thus, assuming that it is infeasible for a PPT oracle algorithm to solve $\ell$ discrete logarithm instances at the same time when restricted to making a *single* query to the discrete logarithm solver oracle, the standalone Schnorr protocol is witness hiding. Similar results can also be obtained for the Guillou-Quisquater's protocol and some other $\Sigma$-protocols for group homomorphisms.

Our reduction $R$ leads to the following *tailored instance compression* problem for DL: Construct a triplet of efficient algorithms $(\mathsf{Z}, \mathsf{C}, \mathsf{U})$ such that: On input $\ell$ instances $(y_1, ..., y_\ell)$ of DL, the compression algorithm $\mathsf{Z}$ outputs a single DL instance $y$; on input $(y_1, ..., y_\ell)$ together with their corresponding witnesses $(w_1, ..., w_\ell)$, the witness compression algorithm $\mathsf{C}$[2] outputs a witness $w$ to the instance $y \leftarrow \mathsf{Z}(y_1, ..., y_\ell)$; given the witness $w$ to $y$, the unfolding algorithm $\mathsf{U}$ outputs all witnesses $(w_1, ..., w_\ell)$ to these $\ell$ instances.

Note that if there exists a successful malicious verifier $V$, then our reduction $R$ together with $V$ can be used to construct a good instance compression scheme for DL problem. Thus, our result on Schnorr protocol can be rephrased as follows: If the tailored instance compression scheme for DL does not exist, then Schnorr protocol is secure.

*What if instance compression schemes exist for DL and RSA?* We observe that the existence of instance compression scheme for DL/RSA with strong parameters has somewhat surprising consequences.

The first consequence is that, assuming the existence of good instance compression scheme for DL, then for any polynomial $\ell$, the AND-DL statement $\{(y_1, y_2, \ldots, y_\ell, g, \mathbb{G}) : \exists w_1, w_2, \ldots, w_\ell, \text{ s.t. } \wedge_{i=1}^{\ell} g^{w_i} = y_i\}$ admits a zero knowledge proof with extremely efficient communication of size $O(1)$ group elements.

[2] It is easy to see that we can construct the witness compression algorithm $\mathsf{C}$ by making simple adaptation to the compressing part of our embedding reduction.

The existence of tailored instance compression scheme for RSA yields a similar consequence.

The second consequence is a construction of non-trivial hash combiner for hash functions based on DL problem. Recall that given a group $\mathbb{G}$, its generator $g$ and a random element $y \in \mathbb{G}$, we have a hash function $H_{(g,y)} : (m_0, m_1) \rightarrow g^{m_0} y^{m_1}$ that is collision-resistant. The hash combiner for DL-based hash functions is of particular interest in the scenario where a set of mutually untrusting parties, given a group $\mathbb{G}$ and $g$, want to set up a single collision-resistant hash function trusted by every one.

Several previous papers [6,26,27] defined *universal* hash combiners (that works for arbitrary hash functions), and showed non-trivial fully black-box combiners do not exist. Note that the above hash combiner needs to take the common parameters of the group and its generator, and works only for DL-based hash functions. However, it is still inconceivable that the above hash combiner with large $\ell$ exists in the real world.

We view these strong consequences as positive evidences for the security of Schnorr and Guillou-Quisquater protocols.

### 1.3 Comparison with a Concurrent Work

In a very recent concurrent work [20], Jain et al. develop a new exciting simulation strategy and construct 2/3-round witness hiding protocol based on some standard number theoretic assumptions for *all* unique witness NP-relations. Our Input-distribution-switching technique gives only witness hiding for *some* cryptographic unique witness relations, however, it applies to existing classic protocols, which are much more efficient and require weaker assumptions[3] than the constructions of [20]. Furthermore, these classic protocols are all public-coin, and such a property usually makes them more vesertile and applicable.

## 2 Preliminaries

Due to space limitations, we refer readers to [13,21] for formal definitions of basic notions and primitives. Here we give only definitions of witness indistinguishable and witness hiding protocols.

**Interactive Proofs.** An *interactive proof system* $\langle P, V \rangle$ [15] for a language $L$ is a pair of interactive Turing machines in which the prover $P$ wishes to convince the verifier $V$ of some statement $x \in L$. We denote by $\langle P, V \rangle(x)$ the output of $V$ at the end of interaction on common input $x$, and without loss of generality, we have the verifier $V$ outputs 1 (resp. 0) if $V$ accepts (resp. rejects).

**Definition 1 (Interactive Proofs).** *A pair of interactive Turing machines $\langle P, V \rangle$ is called an interactive proof system for language $L$ if $V$ is a PPT machine and the following conditions hold:*

[3] Note that the 3-round Blum protocol and GMW protocol can be constructed from one way permutations.

- *Completeness: For every $x \in L$,* $\Pr[\langle P, V\rangle(x) = 1] = 1$.
- *Soundness: For every $x \notin L$, and every (unbounded) prover $P^*$, there exists a negligible function $\mu(n)$ (where $|x| = n$) such that*

$$\Pr[\langle P^*, V\rangle(x) = 1] < \mu(n).$$

An interactive *argument* [4] is an interactive proof except that for which soundness is only required to hold against PPT cheating provers. We often use "protocol" to refer to both proof system and argument system.

**Witness Indistinguishability.** Witness indistinguishable proof system guarantees that if the statement has two independent witnesses, then the malicious verifier cannot tell which witness is being used by the prover in an execution of the protocol.

**Definition 2 (Witness Indistinguishability).** *Let $L$ be an NP language defined by $R_L$. We say that $\langle P, V\rangle$ is witness indistinguishable for relation $R_L$ if for every PPT $V^*$ and every sequence $\{(x, w, w')\}_{x\in L}$, where $(x, w), (x, w') \in R_L$ the following two probability ensembles are computationally indistinguishable:*

$$\{\langle P(w), V^*\rangle(x)\}_{x\in L} \overset{c}{\approx} \{\langle P(w'), V^*\rangle(x)\}_{x\in L}.$$

**Witness Hiding.** Loosely speaking, witness hiding of a protocol [12] refers to the following property: for an input $x \in L$ that is being proven, if a verifier can extract a witness in $R_L(x)$ after interacting with the prover, then he could have done so without such an interaction. This notion is formally defined with respect to a distribution ensemble over inputs as follows.

**Definition 3 (Distribution of Hard Instances).** *Let $L$ be an NP language defined by $R_L$. Let $\mathcal{X} = \{X_n\}_{n\in\mathbb{N}}$ be a distribution ensemble. We say that $\mathcal{X}$ is hard for $R_L$ if for every PPT machine $M$*

$$\Pr\left[M(X_n) \in R_L(X_n)\right] < \mu(n).$$

**Definition 4 (Witness Hiding (under Sequential Repetition)).** *Let $L$ be an NP language defined by $R_L$, $(\mathcal{X}, \mathcal{W}) = \{(X_n, W_n)\}_{n\in\mathbb{N}}$ be a distribution over $R_L$. We say $\langle P, V\rangle$ is witness hiding for $(\mathcal{X}, \mathcal{W})$ if for every PPT machine $V^*$*

$$\Pr\left[\langle P(W_n), V^*\rangle(X_n) \in R_L(X_n)\right] < \mu(n).$$

*We say that $\langle P, V\rangle$ is witness hiding under sequential repetition if it is witness hiding for $(\mathcal{X}, \mathcal{W})$ under any polynomially number of sequential repetitions.*

*Remark 1.* According to our definition of witness hiding, it is easy to verify that if there is witness hiding protocol for $(\mathcal{X}, \mathcal{W})$, then the distribution ensemble $\mathcal{X} = \{X_n\}_{n\in\mathbb{N}}$ on instances must be hard.

# 3 Witness Hiding Protocols for Hard Distributions with Unique Witnesses

In this section we prove a general theorem on witness hiding of constant-round public-coin proofs systems for unique witness relations and present its applications to several cryptographic problems.

## 3.1 A General Theorem

Let $L_1$ and $L_2$ be NP languages (possibly the same), $R_{L_1}$ and $R_{L_2}$ be their corresponding witness relations. Let $(\mathcal{X}^1, \mathcal{W}^1) = \{(X_n^1, W_n^1)\}_{n \in \mathbb{N}}$ be a distribution ensemble over $R_{L_1}$ with *unique* witnesses, and $(\mathcal{X}^2, \mathcal{W}^2) = \{(X_n^2, W_n^2)\}_{n \in \mathbb{N}}$ be a distribution ensemble over $R_{L_2}$ with *multiple* witnesses.

**Theorem 1.** *If the above distribution ensembles satisfy the following conditions:*

1. *$(\mathcal{X}^1, \mathcal{W}^1)$ and $(\mathcal{X}^2, \mathcal{W}^2)$ are computationally indistinguishable.*
2. *For every PPT machine $M$, there is negligible function $\mu(n)$, such that*

$$\Pr\left[(x, w) \leftarrow (X_n^2, W_n^2); w' \leftarrow M(x, w) : w' \in R_L(x) \wedge w \neq w'\right] < \mu(n).$$

3. *For every $n$ and $x$ in $X_n^2$, witnesses in $R_{L_2}(x)$ are uniformly distributed.*[4]

*Then,* any *witness indistinguishable proof systems (including the parallelized version of 3-round public-coin proofs of* [3,14] *and ZAPs of* [11,16]*) are witness hiding (under sequential repetition) for $(\mathcal{X}^1, \mathcal{W}^1)$.*

*Proof.* Let $\langle P, V \rangle$ be an arbitrary witness indistinguishable proof system. In the following, we present our proof only for the standalone case. Note that the same proof works also for these protocols under sequential repetition.

Suppose, towards a contradiction, that there are infinitely many $n$, a polynomial $p$, and a PPT verifier $V^*$ such that

$$\Pr\left[\left\langle P(W_n^1), V^* \right\rangle (X_n^1) \in R_{L_1}(X_n^1)\right] > \frac{1}{p(n)}. \tag{1}$$

Let $\mathbb{S}$ be the set of such $n$'s. Fix an $n \in \mathbb{S}$ and consider the following two experiments:

**EXP**$^b$ ($b \in \{1, 2\}$): Sample $(x, w) \leftarrow (X_n^b, W_n^b)$, play the role of honest prover $P(x, w)$ and interact with $V^*(x)$. When $V^*$ terminates, output what $V^*$ outputs.

Denote by WIN$^b$ that EXP$^b$ outputs a witness for $x$. By the indistinguishability of $(\mathcal{X}^1, \mathcal{W}^1)$ and $(\mathcal{X}^2, \mathcal{W}^2)$, we have the following claim (we shall turn to detailed proof shortly) for some negligible function $\mu(n)$:

[4] This condition can be significantly relaxed, but we stick to it for simplifying presentation.

**Claim 1.** The probabilility $\Pr[\mathrm{WIN}^2]$ is negligibly close to $\frac{1}{p(n)}$, i.e.,

$$\Pr\left[\mathrm{WIN}^2\right] = \Pr\left[\left\langle P(W_n^2), V^*\right\rangle (X_n^2) \in R_{L_2}(X_n^2)\right] > \frac{1}{p(n)} - \mu(n). \qquad (2)$$

It follows from the second property of $(X_n^2, W_n^2)$ that

$$\Pr\left[(x,w) \leftarrow (X_n^2, W_n^2) : \langle P(w), V^*\rangle (x) = w' \in R_{L_2}(x) \wedge w' \neq w\right] < \mu(n). \quad (3)$$

Now by (2) and (3), we have

$$\Pr\left[(x,w) \leftarrow (X_n^2, W_n^2) : \langle P(w), V^*\rangle (x) = w' \wedge w' = w\right] > \frac{1}{p(n)} - \mu(n). \quad (4)$$

which can be rewritten as

$$\begin{aligned} &\Pr\left[(x,w) \leftarrow (X_n^2, W_n^2) : \langle P(w), V^*\rangle (x) = w' \wedge w' = w\right] \\ = &\sum_w \sum_x \Pr\left[\langle P(w), V^*\rangle (x) = w' \wedge w' = w\right] \Pr\left[w \leftarrow W_n^2 | x\right] \Pr\left[x \leftarrow X_n^2\right] \\ > &\frac{1}{p(n)} - \mu(n). \end{aligned}$$

Theorem 1 follows from the following two claims.

**Claim 2.** There exists $x$ in the support of $X_n^2$ satisfying the following two conditions:

- $\sum_w \Pr\left[\langle P(w), V^*\rangle (x) = w' \wedge w' = w\right] \Pr\left[w \leftarrow W_n^2 | x\right] > \frac{1}{2p(n)} - \mu(n).$
- $\sum_w \Pr\left[\langle P(w), V^*\rangle (x) = w' \in R_{L_2}(x) \wedge w' \neq w\right] \Pr\left[w \leftarrow W_n^2 | x\right] < \mu(n).$

**Claim 3.** There exists $x$ in the support of $X_n^2$, $w_1, w_2 \in R_{L_2}(x)$ such that

$$|\Pr\left[\langle P(w_1), V^*\rangle (x) = w_1\right] - \Pr\left[\langle P(w_2), V^*\rangle (x) = w_1\right]| > \frac{1}{\mathsf{poly}(n)}.$$

Note that Claim 3 holds for each $n \in \mathbb{S}$, and thus we conclude that $V^*$ breaks the witness indistinguishability of $\langle P, V\rangle$ on a sequence $\{(x, w_1, w_2)\}_{x \in X_n^2, n \in \mathbb{S}}$, which contradicts the fact that $\langle P, V\rangle$ is witness indistinguishable for *multiple* witnesses relation. This proves Theorem 1. □

We now give the detailed proofs of the above three claims.

*Proof (of **Claim 1**).* Let $p_1(n) = \frac{1}{p(n)}$ (as in (1)), and

$$p_2(n) = \Pr\left[\mathrm{WIN}^2\right] = \Pr\left[\left\langle P(W_n^2), V^*\right\rangle (X_n^2) \in R_{L_2}(X_n^2)\right].$$

Suppose toward a contradiction that $p_1 - p_2 > 1/\mathsf{poly}(n)$. (w.l.o.g., and we assume $p_1 > p_2$.) Consider the following $D$ for distinguishing $(X_n^1, W_n^1)$ and

$(X_n^2, W_n^2)$: Given a sample $(x, w)$ from $(X_n^b, W_n^b)$ (for unknown $b$), $D$ plays the role of honest prover $P(x, w)$ and interact with $V^*(x)$. When $V^*$ terminates, output 1 if the output of $V^*$ is in $R_{L_1}(x)$[5] and 0 otherwise.

Observe that,

$$\begin{aligned}
&\Pr[D(X_n^1, W_n^1) = 1] - \Pr[D(X_n^2, W_n^2) = 1] \\
&\quad = \Pr[\langle P(W_n^1), V^* \rangle (X_n^1) \in R_{L_1}(X_n^1)] - \Pr[\langle P(W_n^2), V^* \rangle (X_n^2) \in R_{L_1}(X_n^2)] \\
&\quad = p_1 - \Pr[\langle P(W_n^2), V^* \rangle (X_n^2) \in R_{L_1}(X_n^2) \wedge \langle P(W_n^2), V^* \rangle (X_n^2) \in R_{L_2}(X_n^2)] \\
&\qquad - \Pr[\langle P(W_n^2), V^* \rangle (X_n^2) \in R_{L_1}(X_n^2) \wedge \langle P(W_n^2), V^* \rangle (X_n^2) \notin R_{L_2}(X_n^2)] \\
&\quad > p_1 - p_2 - \Pr[\langle P(W_n^2), V^* \rangle (X_n^2) \in R_{L_1}(X_n^2) \wedge \langle P(W_n^2), V^* \rangle (X_n^2) \notin R_{L_2}(X_n^2)].
\end{aligned}$$

Now if the last term

$$p_3(n) = \Pr[\langle P(W_n^2), V^* \rangle (X_n^2) \in R_{L_1}(X_n^2) \wedge \langle P(W_n^2), V^* \rangle (X_n^2) \notin R_{L_2}(X_n^2)]$$

is negligible, we conclude that $D$ distinguishes $(X_n^1, W_n^1)$ and $(X_n^2, W_n^2)$, contradicting our assumption. Now we show $p_3(n)$ is negligible. For simplicity, denote by $optV^*(x)$ the output of $V^*$ after interaction with the prover, and we have

$$\begin{aligned}
p_4(n) &= \Pr[\langle P(W_n^1), V^* \rangle (X_n^1) \in R_{L_1}(X_n^1) \wedge \langle P(W_n^1), V^* \rangle (X_n^1) \notin R_{L_2}(X_n^1)] \\
&= \Pr[(x, w) \leftarrow (X_n^1, W_n^1) : optV^*(x) \in R_{L_1}(x) \wedge optV^*(x) \notin R_{L_2}(x)] \\
&\leq \Pr[(x, w) \leftarrow (X_n^1, W_n^1) : w \in R_{L_1}(x) \wedge w \notin R_{L_2}(x)].
\end{aligned}$$

The last equation follows from the uniqueness of $R_{L_1}(x)$ (that is, the valid witness output by $V^*$ in $R_{L_1}(x)$ must be $w$). Observe that $p_4$ must be negligible since otherwise $R_{L_2}$ will serve as a distinguisher that can distinguish $(X_n^1, W_n^1)$ and $(X_n^2, W_n^2)$.

It follows that $p_3$ is negligible either, since otherwise we will have that $|p_3 - p_4|$ is non-negligible, and this leads to the following distinguisher $D'$: Act in the same way as $D$, except that $D'$ output 1 if the output of $V^*$ is in $R_{L_1}(x)$ but not in $R_{L_2}(x)$. It is easy to verify that $D'$ can distinguish $(X_n^1, W_n^1)$ and $(X_n^2, W_n^2)$ with non-negligible probability. □

We now turn to the proof of Claim 2.

*Proof (of **Claim 2**).* We define the following two random events conditioned on a given fixed pair $(x, w)$:

- $\text{EVENT}_{eq}|_{(x,w)}$: $\langle P(w), V^* \rangle (x) = w' \wedge w' = w$;
- $\text{EVENT}_{neq}|_{(x,w)}$: $\langle P(w), V^* \rangle (x) = w' \in R_{L_2}(x) \wedge w' \neq w$,

[5] Note that here we always use $R_{L_1}$ as the tester.

where both events take over the randomnesses used by $P$ and $V^*$. Define the following two sets:

- $\mathbb{H}$: $\{x : \sum_w \Pr\left[\mathrm{EVENT}_{eq}|_{(x,w)}\right] \Pr\left[w \leftarrow W_n^2|x\right] > \frac{1}{2p(n)} - \mu(n)\}$.
- $\mathbb{K}$: $\{x : \sum_w \Pr\left[\mathrm{EVENT}_{neq}|_{(x,w)}\right] \Pr\left[w \leftarrow W_n^2|x\right] < \mu(n)\}$.

Observe that

$$\begin{aligned}
&\frac{1}{p(n)} - \mu(n) < \Pr\left[(x,w) \leftarrow (X_n^2, W_n^2) : \left\langle P(w), V^* \right\rangle (x) = w' \wedge w' = w\right] \\
&\quad = \sum_w \sum_{x \in \mathbb{H}} \Pr\left[\mathrm{EVENT}_{eq}|_{(x,w)}\right] \Pr\left[w \leftarrow W_n^2|x\right] \Pr\left[x \leftarrow X_n^2\right] \\
&\qquad + \sum_w \sum_{x \notin \mathbb{H}} \Pr\left[\mathrm{EVENT}_{eq}|_{(x,w)}\right] \Pr\left[w \leftarrow W_n^2|x\right] \Pr\left[x \leftarrow X_n^2\right] \\
&\quad = \sum_w \Pr\left[\mathrm{EVENT}_{eq}|_{(x,w)}\right] \Pr\left[w \leftarrow W_n^2|x \in \mathbb{H}\right] \Pr\left[x \leftarrow X_n^2 : x \in \mathbb{H}\right] \\
&\qquad + \sum_w \Pr\left[\mathrm{EVENT}_{eq}|_{(x,w)}\right] \Pr\left[w \leftarrow W_n^2|x \notin \mathbb{H}\right] \Pr\left[x \leftarrow X_n^2 : x \notin \mathbb{H}\right],
\end{aligned}$$

which, by the definitions of $\mathrm{EVENT}_{eq}$ and set $\mathbb{H}$, leads to

$$\Pr\left[x \leftarrow X_n^2 : x \in \mathbb{H}\right] > \frac{1}{2p(n)} - \mu(n). \tag{5}$$

Similarly, by (3), we have

$$\begin{aligned}
&\mu(n) > \Pr\left[(x,w) \leftarrow (X_n^2, W_n^2) : \left\langle P(w), V^* \right\rangle (x) = w' \in R_{L_2}(x) \wedge w' \neq w\right] \\
&\quad = \sum_w \sum_{x \in \mathbb{K}} \Pr\left[\mathrm{EVENT}_{neq}|_{(x,w)}\right] \Pr\left[w \leftarrow W_n^2|x\right] \Pr\left[x \leftarrow X_n^2\right] \\
&\qquad + \sum_w \sum_{x \notin \mathbb{K}} \Pr\left[\mathrm{EVENT}_{neq}|_{(x,w)}\right] \Pr\left[w \leftarrow W_n^2|x\right] \Pr\left[x \leftarrow X_n^2\right] \\
&\quad = \sum_w \Pr\left[\mathrm{EVENT}_{neq}|_{(x,w)}\right] \Pr\left[w \leftarrow W_n^2|x \in \mathbb{K}\right] \Pr\left[x \leftarrow X_n^2 : x \in \mathbb{K}\right] \\
&\qquad + \sum_w \Pr\left[\mathrm{EVENT}_{neq}|_{(x,w)}\right] \Pr\left[w \leftarrow W_n^2|x \notin \mathbb{K}\right] \Pr\left[x \leftarrow X_n^2 : x \notin \mathbb{K}\right],
\end{aligned}$$

which, by the definitions of $\mathrm{EVENT}_{neq}$ and set $\mathbb{K}$, leads to

$$\Pr\left[x \leftarrow X_n^2 : x \in \mathbb{K}\right] > 1 - \mu'(n) \tag{6}$$

for some negligible function $\mu'(n)$.

Thus, by (5) and (6), we conclude

$$\Pr\left[x \leftarrow X_n^2 : x \in \mathbb{H} \cap \mathbb{K}\right] > \frac{1}{2p(n)} - \mu(n) - \mu'(n),$$

which means there exist at least one $x$ in the support of $X_n^2$ that satisfies both conditions of Claim 2, as desired. □

The proof of Claim 3 is based on Claim 2.

*Proof (of* ***Claim 3****).* Fix a $x$ in the support of $X_n^2$ that satisfies the two conditions of Claim 1. Note that $W_n^2$ is uniformly distributed on $R_{L_2}(x)$, and by the first condition of Claim 2, we have a $w_1 \in R_{L_2}(x)$ such that

$$\Pr\left[\langle P(w_1), V^*\rangle(x) = w_1\right] > \frac{1}{2p(n)} - \mu(n).$$

By the second condition of Claim 2, we can obtain another witness $w_2 \in R_{L_2}(x)$, $w_2 \neq w_1$, such that

$$\Pr\left[\langle P(w_2), V^*\rangle(x) = w_1\right] < \mu(n),$$

since otherwise, we would have

$$\begin{aligned}
&\sum_{w} \Pr\left[\langle P(w), V^*\rangle(x) = w' \in R_{L_2}(x) \wedge w' \neq w\right] \Pr\left[w \leftarrow W_n^2 | x\right] \\
&\geqslant \sum_{w_2 (\neq w_1)} \Pr\left[\langle P(w_2), V^*\rangle(x) = w_1\right] \Pr\left[w_2 \leftarrow W_n^2 | x : w_2 \neq w_1\right] \\
&= \sum_{w_2 (\neq w_1)} \Pr\left[\langle P(w_2), V^*\rangle(x) = w_1\right] \frac{|R_{L_2}(x)| - 1}{|R_{L_2}(x)|} \\
&> \frac{1}{\mathsf{poly}(n)} \cdot \frac{|R_{L_2}(x)| - 1}{|R_{L_2}(x)|},
\end{aligned}$$

which breaks the second condition of Claim 2[6]. Thus we obtain a desired tuple $(x, w_1, w_2)$, completing the proof of Claim 3. □

### 3.2 Examples of Distributions on Unique Witness Relations

In this subsection, we present several examples of distributions $(\mathcal{X}^1, \mathcal{W}^1)$ on hard unique witness relations that have coupled distributions (satisfing the "if conditions" of Theorem 1), including distributions over OR-DDH tuples with unique witnesses, the images of lossy trapdoor functions and commitments with unique openings. Thus, for these distributions on unique witness relations, the classic constant-round public-coin proof systems, such as parallelized version of classic 3-round public-coin proofs of [3,14] and ZAPs of [11,16], are witness hiding.

**Example 1: OR-DDH Tuples with Unique Witnesses.** The first example is for distribution $(\mathcal{X}^1, \mathcal{W}^1)$ on hard instances with unique witnesses based on DDH assumption.

[6] Note that $|R_{L_2}(x)| > 1$.

**DDH assumption:** Let $\mathsf{Gen}$ be a randomized algorithm that on security parameter $n$ outputs $(\mathbb{G}, g, q)$, where $\mathbb{G}$ is a cyclic group of order $q$ with generator $g$. Then for a randomly chosen triplet $(a, b, c)$, for every PPT algorithm $\mathcal{A}$, there exists a negligible function $\mu(n)$ such that

$$|\Pr[\mathcal{A}((\mathbb{G}, g, q), g^a, g^b, g^{ab}) = 1] - \Pr[\mathcal{A}((\mathbb{G}, g, q), g^a, g^b, g^c) = 1]| < \mu(n).$$

Now, we consider the following two distribution ensembles $(\mathcal{X}^1, \mathcal{W}^1) = \{(X_n^1, W_n^1)\}_{n\in\mathbb{N}}$ and $(\mathcal{X}^2, \mathcal{W}^2) = \{(X_n^2, W_n^2)\}_{n\in\mathbb{N}}$ based on the DDH assumption:

- $(X_n^1, W_n^1) = \{((\mathbb{G}, g, q), x, w) : (\mathbb{G}, g, q) \leftarrow \mathsf{Gen}(1^n)$, the instance $x$ is an OR-DDH tuples $(g^{a_1}, g^{a_2}, g^{a_1 a_2})$ or $(g^{b_1}, g^{b_2}, g^c)$ (where $c \neq b_1 b_2$) with the unique witness $w = (a_1, a_2, a_1 a_2)\}$;
- $(X_n^2, W_n^2) = \{((\mathbb{G}, g, q), x, w) : (\mathbb{G}, g, q) \leftarrow \mathsf{Gen}(1^n)$, the instance $x$ is an OR-DDH tuples $(g^{a_1}, g^{a_2}, g^{a_1 a_2})$ or $(g^{b_1}, g^{b_2}, g^{b_1 b_2})$ with multiple witnesses $w_0 = (a_1, a_2, a_1 a_2)$, $w_1 = (b_1, b_2, b_1 b_2)\}$.

Based on Theorem 1, we have that all the witness hiding protocols for $(\mathcal{X}^2, \mathcal{W}^2)$ above are also witness hiding for $(\mathcal{X}^1, \mathcal{W}^1)$ above, under the DDH assumption.

**Example 2: Lossy Trapdoor Functions.** We now present another example of distribution ensembles $(\mathcal{X}^1, \mathcal{W}^1)$ based on lossy trapdoor functions.

Recall the definition of lossy trapdoor functions [25]. Let $n$ be the security parameter (representing the input length of the function) and $\ell(n)$ be the lossiness of the collection.

**Definition 5.** *A collection of $(m, k)$-lossy trapdoor functions is given by a tuple of PPT algorithms $(\mathsf{Gen}, \mathsf{F}, \mathsf{F}^{-1})$. It satisfies the following property:*

- *Easy to sample an injective function with trapdoor: $\mathsf{Gen}_{inj}(\cdot) := \mathsf{Gen}(\cdot, 1)$ outputs $(s, t)$ where $s$ is the description of an injective function $f_s$ and $t$ is its trapdoor, $\mathsf{F}(s, \cdot)$ computes the function $f_s(\cdot)$ over the domain $\{0, 1\}^n$, and $\mathsf{F}(t, \cdot)$ computes the function $f_s^{-1}(\cdot)$. If a value $y$ is not in the image of $f_s$, then $\mathsf{F}(t, y)$ is unspecified.*
- *Easy to sample a lossy function: $\mathsf{Gen}_{lossy}(\cdot) := \mathsf{Gen}(\cdot, 0)$ outputs $(s, \perp)$ where $s$ is the description of function $f_s$, and $\mathsf{F}(s, \cdot)$ computes the function $f_s(\cdot)$ over the domain $\{0, 1\}^m$ whose image has size at most $2^{m-k}$.*
- *Hard to distinguish injective and lossy: the first outputs of $\mathsf{Gen}_{inj}$ and $\mathsf{Gen}_{lossy}$ are computationally indistinguishable.*

Now we consider the following two distribution ensembles $(\mathcal{X}^1, \mathcal{W}^1) = \{(X_n^1, W_n^1)\}_{n\in\mathbb{N}}$ and $(\mathcal{X}^2, \mathcal{W}^2) = \{(X_n^2, W_n^2)\}_{n\in\mathbb{N}}$ based on lossy trapdoor function:

- $(X_n^1, W_n^1) := \{((s, y), w) : s \leftarrow \mathsf{Gen}_{\mathrm{inj}}(1^n); w \leftarrow \{0, 1\}^n; f_s(w) = y\}$.
- $(X_n^2, W_n^2) := \{((s, y), w) : s \leftarrow \mathsf{Gen}_{\mathrm{lossy}}(1^n); w \leftarrow \{0, 1\}^n; f_s(w) = y\}$.

Note that the description of a lossy function is indistinguishable from that of an injective function, thus the distribution $(X_n^2, W_n^2)$ over the description of lossy function together with its input-output pair is also indistinguishable from the distribution $(X_n^1, W_n^1)$ over injective function together with its input-output pair, since otherwise if we have a PPT $D'$ that can distinguish $(X_n^1, W_n^1)$ from $(X_n^2, W_n^2)$, we will have a PPT $D$ that can tell apart lossy functions from injective ones: When being given a description of a function $f$, $D$ samples input w and computes $y = f(w)$ and then invokes $D'$ on $(f, y, w)$ and outputs what $D'$ outputs.

It is also easy to verify (using the fact that there is only a single $w$ such that $f(w) = y$ for a fixed injective function $f$ and $y$.)that the second condition of Theorem 1 holds. When sampling $w$ in the domain of a lossy function $f$ uniformly, then for a fixed output $y$, those pre-images of $y$ are uniformly distributed over $\{w : f(w) = y\}$. Hence, the above two distributions satisfy the third condition of Theorem 1.

Thus, it follows from Theorem 1 that all the witness hiding protocols for $(\mathcal{X}^2, \mathcal{W}^2)$ above are also witness hiding for $(\mathcal{X}^1, \mathcal{W}^1)$ above, under the existence of lossy trapdoor functions.

**Example 3: Commitments with Unique Openings.** Our third example of distribution ensembles $(\mathcal{X}^1, \mathcal{W}^1)$ is based on mixed commitments [9,16].

A mixed commitment scheme is basically a commitment scheme that has two different flavors of key generation algorithms. In the binding mode, $\mathsf{Gen}_1$ generates a perfectly binding commitment key, in which case a valid commitment uniquely defines one possible message. In the hiding mode, $\mathsf{Gen}_2$ generates a perfectly hiding commitment key, in which case the commitment reveals no information whatsoever about the message. Moreover, two kinds of keys are computationally indistinguishable.

Now, we consider the following two distribution ensembles $(\mathcal{X}^1, \mathcal{W}^1) = \{(X_n^1, W_n^1)\}_{n\in\mathbb{N}}$ and $(\mathcal{X}^2, \mathcal{W}^2) = \{(X_n^2, W_n^2)\}_{n\in\mathbb{N}}$ based on the mixed commitments:

- $(X_n^1, W_n^1) = \{((x, pk), (m, r)) : pk \leftarrow \mathsf{Gen}_1(1^n); m \overset{\mathrm{R}}{\leftarrow} M; r \overset{\mathrm{R}}{\leftarrow} R; x \leftarrow \mathsf{Com}_{pk}(m; r)\}$.
- $(X_n^2, W_n^2) = \{((x, pk), (m, r)) : pk \leftarrow \mathsf{Gen}_2(1^n); m \overset{\mathrm{R}}{\leftarrow} M; r \overset{\mathrm{R}}{\leftarrow} R; x \leftarrow \mathsf{Com}_{pk}(m; r)\}$.

Assuming the existence of mixed commitments, we can use the reasoning similar to the case of lossy functions and conclude that all the witness hiding protocols for $(\mathcal{X}^2, \mathcal{W}^2)$ above are also witness hiding for $(\mathcal{X}^1, \mathcal{W}^1)$ above.

## 4 Embedding Reduction: The Security of Schnorr and Guillou-Quisquater Protocols and Instance Compression

In this section, we develop an embedding reduction technique to base the witness hiding security[7] of Schnorr protocol on non-existence of tailored instance compression scheme for discrete logarithm.

Similar results can also be obtained for the Guillou-Quisquater's protocol and some other $\Sigma$-protocols for group homomorphisms. Note that, given a successful adversary $V^*$, our technique yields a tailored instance compression scheme with parameters much stronger than the ones in [2], and thus strengthens the results of [2].

The formal study of instance compression was initiated by Harnik and Naor [19]. We tailor their definition for our purpose. Roughly speaking, a tailored instance compression scheme for a (search) NP problem can compress a long instance(s) into a shorter instance, and given the solution to the shorter instance, we can solve all the original instance(s). It should be noted that the impossibility results of [10] with respect to NP-complete languages also hold for our tailored definition.

**Definition 6 (Tailored Instance Compression for Search Problem).** *Let $L$ be an NP language and $R_L$ its NP relation, and $\mathcal{X} = \{X_n\}_{n\in\mathbb{N}}$ be a distribution ensemble over $L$. A $(\ell(\cdot), \varepsilon(\cdot))$-tailored instance compression scheme for $R_L$ consists of three PPT algorithms* $(\mathsf{Z}, \mathsf{C}, \mathsf{U})$*, such that for sufficiently large $n$:*

- $(x, st) \leftarrow \mathsf{Z}(x_1, \cdots, x_\ell)$*: On input $x_i \in L$ for $i \in [\ell]$, the PPT instances compression algorithm* $\mathsf{Z}$ *outputs a single $x \in L$ and the state $st$.*
- $w \leftarrow \mathsf{C}((x_1, w_1), \cdots, (x_\ell, w_\ell))$*: On input $(x_i, w_i) \in R_L$ for $i \in [\ell]$, the PPT witness compression algorithm* $\mathsf{C}$ *outputs a valid witness $w$ to the instance $x$ generated by* $\mathsf{Z}(x_1, \cdots, x_\ell)$*.*
- $(w_1, \cdots, w_\ell) \leftarrow \mathsf{U}(x, w, st)$*: On input $x \in L$, $st$, together with the corresponding witness $w \in R_L(x)$, the PPT unfolding algorithm* $\mathsf{U}$ *outputs the witnesses $w_i \in R_L(x_i)$ for all $i \in [\ell]$.*
- *For all $w \in R_L(x)$, the following holds:*

$$\Pr\left[\begin{array}{c} (x_1, \ldots, x_\ell) \leftarrow X_n^\ell; \\ (x, st) \leftarrow \mathsf{Z}(x_1, \ldots, x_\ell); \\ (w_1, \ldots, w_\ell) \leftarrow \mathsf{U}(x, w, st); \end{array} : \wedge_{i=1}^{\ell} w_i \in R_L(x_i)\right] > \varepsilon(n)$$

*Remark 2.* Our definition is stronger than the one of [19] in several respects. In the Definition 2.25 of [19], the retrieving algorithm (that corresponds to our witness compression algorithm) does not take witnesses to $(x_1, \ldots, x_\ell)$ as input, and thus is not required to be efficient; the unfolding algorithm above is also not required in [19], but that is the key for our applications of instance compression scheme (if exists).

[7] Note that witness hiding implies the security of identification protocol.

Observe that the one-more like assumptions can be rephrased in the framework of instance compression. For example, the one-more DL assumption is equivalent to assume non-existence of $(\ell, \varepsilon)$-tailored instance compression scheme for DL with weaker requirements: (1) The witness compression algorithm is not required; (2) The instance compression algorithm is allowed to output $\ell - 1$ instances (which leads to much weak compression ratio) and the unfolding algorithm needs to take $\ell - 1$ witnesses correspondingly.

### 4.1 The Security of Schnorr Protocol

Let $\mathbb{G}$ be a cyclic group of order $q$ with the generator $g$, where $q$ is a prime such that $q \mid p - 1$, $p$ is a prime $2^{n-1} \leq p \leq 2^n$. Given a common input $x$, the Schnorr protocol allows the prover $P$ to convince the verifier $V$ of knowledge of the unique discrete logarithm $w$ of $x$ (i.e., $x = g^w$). Formal description of this protocol can be found in Fig. 1.

Given $(g, \mathbb{G})$, we define the NP relation $R_{(g,\mathbb{G})} := \{(x, w) : x = g^w\}$. We show that a successful adversarial verifier will lead to a non-trivial tailored instance compression scheme for discrete logarithm (DL) instances.

**Theorem 2.** *If there exists a PPT algorithm $V^*$ that breaks witness hiding of Schnorr protocol with probability $p$ (i.e. $V^*$ after interaction with the prover $P$ outputs a valid discrete logarithm $w$ of $x$ with probability greater than $p$), then there exists $(\ell, p^{\ell-1})$-tailored instance compression scheme for DL instances in $\mathbb{G}$ for any $\ell$.*

*Remark 3.* It should be noted that for a negligible probability $\varepsilon$, the $(\ell, \varepsilon)$-tailored instance compression scheme (if exists) is barely applicable. For achieving meaningful compression scheme from $V^*$, we should set $\ell$ to be (arbitrary) constant when $p$ is an inverse polynomial; if $p$ is negligibly close to 1, then $\ell$ can be set to be (arbitrary) polynomial. Note also that the technique of [2] gives us only $\ell = 2$.

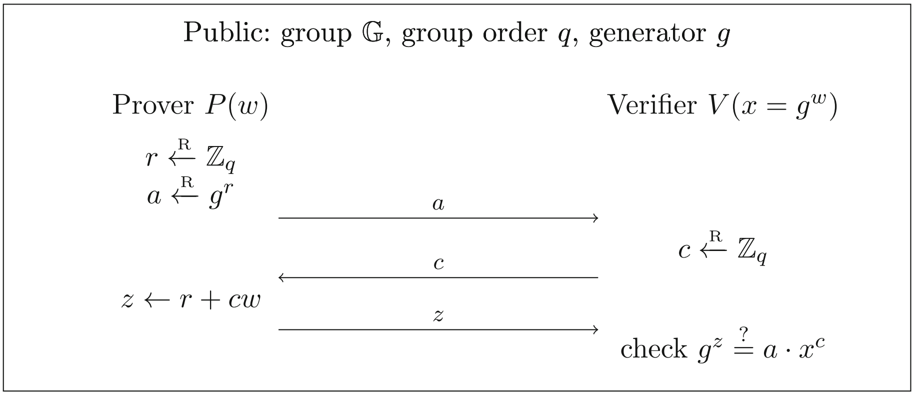
Public: group $\mathbb{G}$, group order $q$, generator $g$
Prover $P(w)$
Verifier $V(x = g^w)$
$r \overset{R}{\leftarrow} \mathbb{Z}_q$
$a \overset{R}{\leftarrow} g^r$
$a$
$c \overset{R}{\leftarrow} \mathbb{Z}_q$
$c$
$z \leftarrow r + cw$
$z$
check $g^z \overset{?}{=} a \cdot x^c$

**Fig. 1.** Schnorr identification scheme

```
$D^{V^*}$
input  : instances $x_1, x_2 \in \mathbb{G}$, random tape $R_V$
Run $V^*$ with random tape $R_V$ on instance $x_1$;
Send $x_2$ as the first prover message to $V^*$;
output: output: If $V^*$ answers with a challenge $c \in \mathbb{Z}_q$, output $x = x_1^c x_2$; else
        output $\perp$.
```

```
$B^{V^*}$
input  : $z \in \mathbb{Z}_q$, $x_1, x_2 \in \mathbb{G}$, random tape $R_V$
Execute the Schnorr protocol with $V^*$ in exactly the same way as $D(x_1, x_2, R_V)$
until receiving the challenge $c$ from $V^*$;
Send $z$, which is supposed to be such that $g^z = {x_1}^c x_2$, to $V^*$;
output: If $V^*$ outputs the witness $w$ satisfying $x_1 = g^w$, output $z_1 = w$ and
        $z_2 = z - cw$; else output $\perp$.
```

```
The compression algorithm $\mathsf{Z}^{V^*}$
input  : $(x_1, x_2, \cdots, x_\ell)$
$st \leftarrow \{x_1, \cdots, x_\ell\}$;
set $x_j^0 = x_j$, for $j = 1, 2, \cdots, \ell$;
for $i \leftarrow 0$ to $l - 1$ do
    for $j \leftarrow 1$ to $2^{l-i-1}$ do
        $x^i_{2j-1} \leftarrow x^i_{2j-1} \cdot g^{r^i_{2j-1}}$, $x^i_{2j} \leftarrow x^i_{2j} \cdot g^{r^i_{2j}}$, where $r^i_{2j-1}, r^i_{2j} \xleftarrow{\mathrm{R}} \mathbb{Z}_q$;
        ${R_V}^i_j \xleftarrow{\mathrm{R}} \{0,1\}^{poly(n)}$, where $poly(n)$ denotes the length of the random
        tape ${R_V}^i_j$;
        $x_j^{i+1} \leftarrow D^{V^*}(x^i_{2j-1}, x^i_{2j}, {R_V}^i_j)$ (if $D$ outputs $\perp$, return $\perp$);
        Add $(x_j^{i+1}, r^i_{2j-1}, r^i_{2j}, {R_V}^i_j)$ to $st$;
    end
end
set $x \leftarrow x_1^l$;
Return $x, st$;
```

We first construct two efficient subroutines $D$ and $B$ for our embedding reduction. On input two instances $(x_1, x_2)$, the algorithm $D$ interacts with $V^*$ (where $x_1$ serves as the common input, and $x_2$ serves as the first prover message) until the challenge $c$ from $V^*$ is received, and outputs a new instance $x_1^c x_2$; on input discrete logarithm $z$ of $x_1^c x_2$, the algorithm $B$ interacts with $V^*$ until the output of $V^*$ is received, and outputs two discrete logarithms of the two instances $(x_1, x_2)$. Formal descriptions of $D$ and $B$ can be found in Algorithm $D^{V^*}$and $B^{V^*}$.

As illustrated in Fig. 2, our embedding black-box reduction naturally corresponds to a pair of efficient algorithms, a compression algorithm $\mathsf{Z}$ and an

```
The unfolding algorithm $\mathsf{U}^{V^*}$
input : $x \in \mathbb{G}$, $w \in \mathbb{Z}_q$, $st$
set $x_1^l \leftarrow x$, $z_1^l \leftarrow w$;
for $i = l-1$ to $0$ do
    for $j = 1$ to $2^{l-i-1}$ do
        Retrieve $x_{2j-1}^i, x_{2j}^i, r_{2j-1}^i, r_{2j}^i$ and ${R_V}_j^i$ from $st$;
        $(z_{2j-1}^i, z_{2j}^i) \leftarrow B^{V^*}(z_j^{i+1}, x_{2j-1}^i, x_{2j}^i, {R_V}_j^i)$ (if $B$ outputs $\perp$, return $\perp$);
        $z_{2j-1}^i \leftarrow z_{2j-1}^i - r_{2j-1}^i$, $z_{2j}^i \leftarrow z_{2j}^i - r_{2j}^i$;
    end
end
output: $(w_1, w_2, \cdots, w_\ell) = (z_1^0, z_2^0, \cdots, z_\ell^0)$
```

unfolding algorithm $\mathsf{U}$. In the first phase, the compression algorithm $\mathsf{Z}$, taking as input discrete logarithm instances $(x_1, \ldots, x_\ell)$, invokes $D$ recursively to generate new instance, each time $D$ transforms two new instances into a new single one. $\mathsf{Z}$ outputs the final single instance $x = x_1^3$ and the corresponding $st$ consisting of all instances input to $D$ and the random tape of $\mathsf{Z}$.

On input a witness $w = z_1^3$ to $x = x_1^3$, the unfolding algorithm $\mathsf{U}$ invokes $B$ recursively, by feeding $B$ with a discrete logarithm of an instance, to solve two instances. Finally, $\mathsf{U}$ will solve all instances $(x_1, x_2, ..., x_\ell)$.

For our analysis to go through, given two instances $x_1, x_2$, the compression algorithm $\mathsf{Z}$ has to choose two random strings $r_1$, $r_2$ and a fresh random tape for $V^*$, and then runs $D$ on input $(x_1 g^{r_1}, x_2 g^{r_2})$. $\mathsf{Z}$ will store all these randomnesses in $st$. The formal descriptions of $\mathsf{Z}$ and $\mathsf{U}$ can be found in Algorithm $D^{V^*}$and $B^{V^*}$respectively. Without loss of generality, we assume that $\ell = 2^l$ for some integer $l$.

*Proof* (of Theorem 1). From Fig. 2, we see the symmetry that, on input two instances $(x_{2j-1}^i, x_{2j}^i)$, $D^{V^*}(x_{2j-1}^i, x_{2j}^i, {R_V}_j^i)$ generates a new instance $x_j^{i+1}$; whereas, on input a discrete logarithm $z_j^{i+1}$ of $x_j^{i+1}$, $B^{V^*}(z_j^{i+1}, x_{2j-1}^i, x_{2j}^i, {R_V}_j^i)$ produces the two discrete logarithms $(z_{2j-1}^i, z_{2j}^i)$ of the two instances $(x_{2j-1}^i, x_{2j}^i)$ that are inputs to $D$.

We say an algorithm wins if it does not output "$\perp$". Note that all these invocations of $D$ are independent, and that, for every $i$,$j$, the $V^*$ success probability $p$ is the probability that both $D^{V^*}(x_{2j-1}^i, x_{2j}^i, {R_V}_j^i)$ and $B^{V^*}(z_j^{i+1}, x_{2j-1}, x_{2j}, {R_V}_j^i)$ win, that is,

$$\Pr[D^{V^*}(x_{2j-1}^i, x_{2j}^i, {R_V}_j^i) \text{ wins } \wedge B^{V^*}(z_j^{i+1}, x_{2j-1}, x_{2j}, {R_V}_j^i) \text{ wins}] = p.$$

Observe that in the entire reduction there are exactly $(\ell - 1)$ pairs of invocations of $D^{V^*}$ and $B^{V^*}$, thus we have the probability

$$\Pr\left[\begin{array}{l} \Pr[(x, st) \leftarrow \mathsf{Z}^{V^*}(x_1, x_2, \cdots, x_\ell); \\ (w_1, w_2, \cdots, w_\ell) \leftarrow \mathsf{U}^{V^*}(x, st, w) \end{array} : \wedge_{i=1}^{\ell} x_i = g^{w_i}\right] = p^{\ell-1}$$

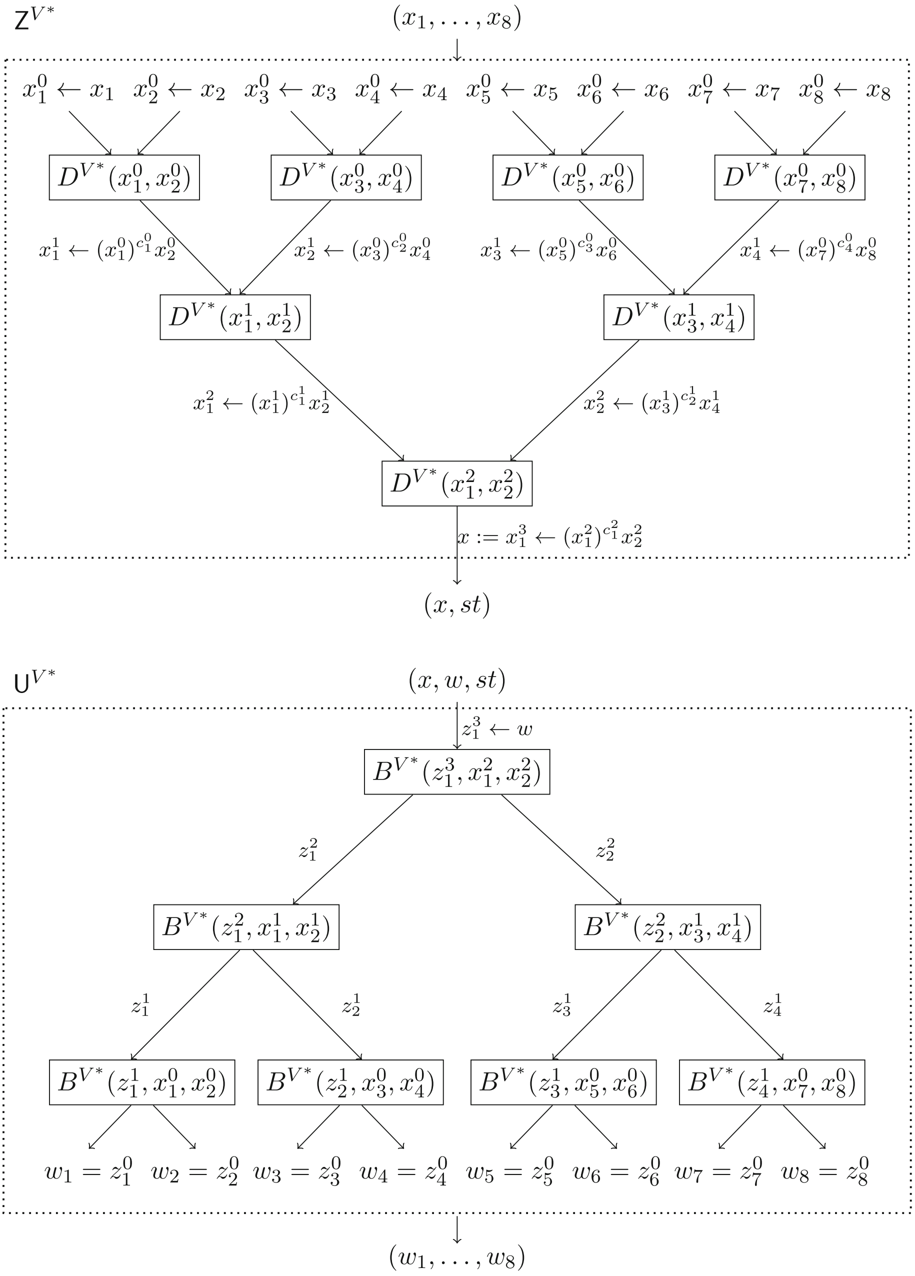
$\mathsf{Z}^{V^*}$
$(x_1, \ldots, x_8)$
$x_1^0 \leftarrow x_1$ $x_2^0 \leftarrow x_2$ $x_3^0 \leftarrow x_3$ $x_4^0 \leftarrow x_4$ $x_5^0 \leftarrow x_5$ $x_6^0 \leftarrow x_6$ $x_7^0 \leftarrow x_7$ $x_8^0 \leftarrow x_8$
$D^{V^*}(x_1^0, x_2^0)$
$D^{V^*}(x_3^0, x_4^0)$
$D^{V^*}(x_5^0, x_6^0)$
$D^{V^*}(x_7^0, x_8^0)$
$x_1^1 \leftarrow (x_1^0)^{c_1^0} x_2^0$
$x_2^1 \leftarrow (x_3^0)^{c_2^0} x_4^0$
$x_3^1 \leftarrow (x_5^0)^{c_3^0} x_6^0$
$x_4^1 \leftarrow (x_7^0)^{c_4^0} x_8^0$
$D^{V^*}(x_1^1, x_2^1)$
$D^{V^*}(x_3^1, x_4^1)$
$x_1^2 \leftarrow (x_1^1)^{c_1^1} x_2^1$
$x_2^2 \leftarrow (x_3^1)^{c_2^1} x_4^1$
$D^{V^*}(x_1^2, x_2^2)$
$x := x_1^3 \leftarrow (x_1^2)^{c_1^2} x_2^2$
$(x, st)$
$\mathsf{U}^{V^*}$
$(x, w, st)$
$z_1^3 \leftarrow w$
$B^{V^*}(z_1^3, x_1^2, x_2^2)$
$z_1^2$
$z_2^2$
$B^{V^*}(z_1^2, x_1^1, x_2^1)$
$B^{V^*}(z_2^2, x_3^1, x_4^1)$
$z_1^1$
$z_2^1$
$z_3^1$
$z_4^1$
$B^{V^*}(z_1^1, x_1^0, x_2^0)$
$B^{V^*}(z_2^1, x_3^0, x_4^0)$
$B^{V^*}(z_3^1, x_5^0, x_6^0)$
$B^{V^*}(z_4^1, x_7^0, x_8^0)$
$w_1 = z_1^0$ $w_2 = z_2^0$ $w_3 = z_3^0$ $w_4 = z_4^0$ $w_5 = z_5^0$ $w_6 = z_6^0$ $w_7 = z_7^0$ $w_8 = z_8^0$
$(w_1, \ldots, w_8)$

**Fig. 2.** Simplified reduction for $\ell = 8$. We assume that $V^*$ is deterministic and with probability 1 it breaks witness hiding of Schnorr protocol.

Note that when given as input all the witnesses $(w_1, \cdots, w_\ell)$ of the target instances $(x_1, \cdots, x_\ell)$ to $\mathsf{Z}$, $\mathsf{Z}$ is able to compute the witness to every instance output by $D$. Thus by making a straightforward adaptation of $\mathsf{Z}$ we get a PPT witness compression algorithm $\mathsf{C}$ as desired. This completes the proof. □

### 4.2 Security of the Guillou-Quisquater Protocol

In this section we state a similar result on Guillou-Quisquater identification protocol [17]. The reduction is essentially the same as the one for Schnorr protocol, and here we omit it.

**The Guillou-Quisquater Protocol.** Let $N = pq$ be an RSA modulus (i.e. $p$ and $q$ are large distinct primes for security parameter $n$) and $e < \phi(N)$ be an odd prime satisfying $\gcd(d, \phi(N)) = 1$ and $ed \equiv 1 \mod \phi(N)$. The Guillou-Quisquater protocol proceeds as follows (See Fig. 3). The prover $P$ wants to convince the verifier $V$ of the unique $e$-th root $w$ modulo $N$ of a given number $x$. First, $P$ chooses $r \in \mathbb{Z}_N^*$ at random and sends $a = r^e \mod N$ to the verifier $V$. Upon receiving the verifier's challenge $c$, $P$ responses with $z = r \cdot w^c$. $V$ accepts if and only if $z^e = a \cdot x^c$.

Given $(e, N)$, we define the NP relation $R_{e,N} := \{(x, w) : x = w^e \bmod N\}$. Similar to the Schnorr protocol, we have the following theorem.

**Theorem 3.** *If there exists a PPT algorithm $V^*$ that breaks witness hiding of Guillou-Quisquater protocol with probability $p$ (i.e. $V^*$ after interaction outputs the witness $w$ with probability greater than $p$), then there exists $(\ell, p^{\ell-1})$-tailored instance compression scheme for RSA instances in $\mathbb{Z}_N^*$ for any $\ell$.*

*Remark 4.* We also note that our reduction can also apply to $\Sigma-$protocols for group homomorphisms [7,22].

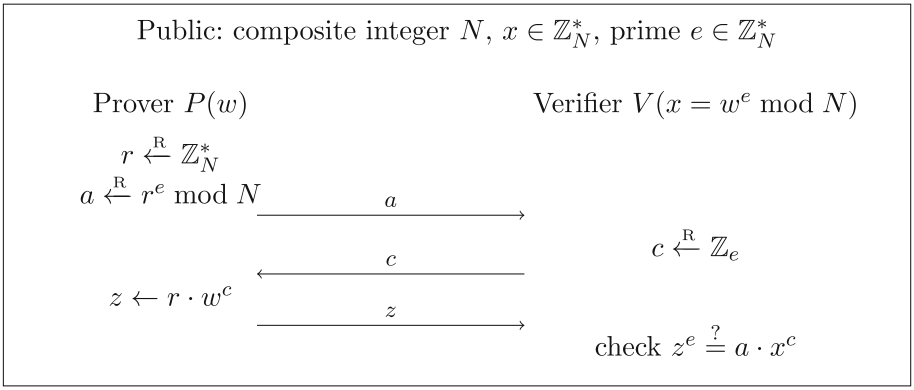
Public: composite integer $N$, $x \in \mathbb{Z}_N^*$, prime $e \in \mathbb{Z}_N^*$
Prover $P(w)$
Verifier $V(x = w^e \bmod N)$
$r \xleftarrow{\text{R}} \mathbb{Z}_N^*$
$a \xleftarrow{\text{R}} r^e \bmod N$
$a$
$c \xleftarrow{\text{R}} \mathbb{Z}_e$
$c$
$z \leftarrow r \cdot w^c$
$z$
check $z^e \overset{?}{=} a \cdot x^c$

**Fig. 3.** GQ identification scheme

# 5 Some Consequences of Existence of Good Tailored Instance Compression Schemes for DL and RSA

In this section, we show some strong consequences of the existence of good tailored instance compression schemes for DL and RSA problems. To simplify our presentation, we consider only $(poly(n), 1 - negl(n))$-tailored instance compression schemes, where $poly(n)$ denotes an arbitrary polynomial in security parameter $n$. Such an instance compression scheme can be constructed from the efficient adversary that can break the witness hiding of Schnorr/Guillou-Quisquater protocol with probability negligibly close to 1. We also stress that, as showed in [24], even for such an adversary, no black-box reduction can turn it into an algorithm that breaks some standard assumptions and reach a contradiction.

## 5.1 Extremely Communication-Efficient Zero Knowledge Protocols for AND-DL and AND-RSA

Suppose that there is a $(\mathsf{poly}(n), 1 - \mathsf{negl}(n))$-tailored instance compression scheme $(\mathsf{Z}, \mathsf{C}, \mathsf{U})$ for DL. In this subsection we further assume that the compression algorithm $\mathsf{Z}$ is deterministic without loss of generality: Since almost all possible random tapes for $\mathsf{Z}$ are good in the sense that on every such random tape $\mathsf{Z}$ will output an instance, together with some state information, for which the unfolding algorithm will succeed, we can publish a good random tape and let each party execute $\mathsf{Z}$ on the same random tape when needed[8].

The immediate consequence of such a tailored instance compression scheme is that, for an arbitrary polynomial $\ell$, the AND-DL statement, $\{(x_1, x_2, \ldots, x_\ell, g, \mathbb{G}) : \exists w_1, w_2 \ldots, w_\ell, s.t. \wedge_{i=1}^{\ell} g^{w_i} = x_i\}$, has a proof of size $|w_i|$, since we can have both the prover and the verifier run $\mathsf{Z}$ on $(x_1, x_2, \ldots, x_\ell)$ and obtain a single instance $x$ of the same size of $x_i$, and then the prover send the $w$ (such that $g^w = x$) to the verifier, which accepts if $g^w = x$ and all $w_i$, obtained from the unfolding algorithm $\mathsf{U}$, satisfy $g^{w_i} = x_i$.

With this succinct proof for the AND-DL statement, the Feige-Shamir zero knowledge protocol of [12] for AND-DL statements can be implemented in an extremely communication-efficient way (with communication of size $O(1)$ group elements).

Protocol Feige-Shamir

Common input: $x_1, x_2, \ldots, x_\ell \in \mathbb{G}$.
The prover $P$'s input: $w_1, w_2, \ldots, w_\ell, s.t. \wedge_{i=1}^{\ell} g^{w_i} = x_i$.

[8] Note that in the last item in Definition 6, the probability takes over the randomness of drawing all the instantces $(x_1, \ldots, x_\ell)$ and the randomnesses of $\mathsf{Z}$ and $\mathsf{U}$. Thus, when $\epsilon(\cdot)$ is set to be $1 - \mathsf{negl}(\cdot)$, by a simple counting argument, we will have a single random tape of $\mathsf{Z}$ (and $\mathsf{U}$) that works for all but negligible fraction of inputs of $\ell$-tuple $(x_1, \ldots, x_\ell)$.

**First phase:** The verifier chooses $w'_0, w'_1 \xleftarrow{\text{R}} \mathbb{Z}_q$ independently and at random, computes $x'_0 = g^{w'_0}$ and $x'_1 = g^{w'_1}$, and then executes the 3-round $\Sigma_{OR}$ protocol (OR-composition of the Schnorr protocol [8]), in which $V$ plays the role of the prover, to prove the knowledge of the witness to the statement $(x'_0 \vee x'_1)$;

**Second phase:** Both the prover and the verifier run $\mathsf{Z}$ on $(x_1, x_2, \ldots, x_\ell)$ and obtain a new instance $x \in \mathbb{G}$, and then the prover runs the witness compression algorithm $\mathsf{C}$ on $w_1, w_2, \ldots, w_\ell$ to obtain $w$ such that $g^w = x$, and proves to the verifier the knowledge of the witness to the statement $(x \vee x'_0 \vee x'_1)$ using $\Sigma_{OR}$ protocol of [8].

This leads to the following proposition.

**Proposition 1.** *If there exists a* $(\mathsf{poly}(n), 1-\mathsf{negl}(n))$*-tailored instance compression scheme for AND-DL, then for an arbitrary polynomial* $\ell(n)$*, the AND-DL statement,* $\{(x_1, x_2, \ldots, x_\ell, g, \mathbb{G}) : \exists w_1, w_2, \ldots, w_\ell, \text{ s.t. } \wedge_{i=1}^{\ell} g^{w_i} = x_i\}$*, has a zero knowledge protocol with communication complexity of* $O(1)$ *group elements.*

### 5.2 Special Hash Combiner

The second consequence is a construction of non-trivial hash combiner for hash functions based on the DL problem, which would help a set of $\ell$ mutually untrusting parties set up a single trusted collision-resistant hash function from a given group.

Consider the cyclic group $\mathbb{G}$ mentioned in Sect. 4.1. Let $x = g^w$ for some $w$. $h^x : {\mathbb{Z}_q}^2 \to \mathbb{G}$ is *collision resistant hash functions* (CRHFs) based on DL problem defined as follows:

$$h^x(m_0, m_1) = g^{m_0} x^{m_1}.$$

Clearly, finding a collision for $h^x$ is equivalent to solving the discrete logarithm problem $w = \log_g x$.

**Definition 7 (Hash Combiner for CRHFs Based on DL Problem).** *A non-uniform PPT Turing machine* $\mathsf{H} : \mathcal{R} \times {\mathbb{Z}_q}^2 \to \{0,1\}^v$ *is said to be a randomized* $(k, \ell)$*-combiner for CRHFs based on DL, if it satisfies the following conditions:*

- *For any given* $\ell$ *elements of* $\mathbb{G}$ *(i.e.* $x_1, \cdots, x_\ell$*), for every* $r \in \mathcal{R}$*,* $\mathsf{H}^{x_1, x_2, \cdots, x_\ell}(r, \cdot, \cdot)$ *is a collision resistant hash function, if at least* $k$ *components* $x_i$ *can be used to construct collision resistant hash functions* $h^{x_i}(\cdot, \cdot)$*.*
- *For every PPT adversary* $\mathcal{B}$ *breaking the collision resistent hash combiner* $\mathsf{H}^{x_1, x_2, \cdots, x_\ell}(r, \cdot, \cdot)$*, there exists a PPT reduction* $R$*, s.t.* $R^{\mathcal{B}}$ *can find collisions for at least* $\ell - k + 1$ *hash functions* $h^{x_i}$*,* $i \in [\ell]$*, with overwhelming probability.*

Now we will show that the combiner for CRHFs based on the DL problem can be constructed by the compression algorithm for DL instances. The previous papers [6,26,27] showed that there doesn't exist "fully"[9] black-box combiners whose output length is significantly smaller than what can be achieved by trivially concatenating the output of any $\ell - k + 1$ of the components. We can construct a special non-black-box $(1, \ell)$-combiner for CRHFs based on DL problem whose output length is significantly smaller using the instances compression algorithm mentioned in Corollary 2, under the discrete logarithm assumption.

**Proposition 2.** *Suppose there exists a* $(\mathsf{poly}(n), 1 - \mathsf{negl}(n))$*-tailored instance compression algorithm for any given* $\ell (= \mathsf{poly}(n))$ *DL instances* $x_1, x_2, \ldots, x_\ell$ *in* $\mathbb{G}$*. Then there exists a randomized* $(1, \ell)$*-combiner* $\mathsf{H}^{x_1, x_2, \ldots, x_\ell}$ *for CRHFs based on DL problem, with the same output length* $v$ *as the regular discrete logarithm hash functions* $h^{x_i}$*.*

*Proof.* Assume that there exists $(\mathsf{poly}(n), 1 - \mathsf{negl}(n))$-tailored instance compression algorithms for DL. That is, for any polynomial $\ell$, there exists a pair of PPT algorithms $(\mathsf{Z}, \mathsf{U})$, for $w = \log_g x$, such that

$$\Pr\left[\begin{array}{c}(x, st) \leftarrow \mathsf{Z}(x_1, \ldots, x_\ell);\\(w_1, \ldots, w_\ell) \leftarrow \mathsf{U}(x, w, st)\end{array} : \wedge_{i=1}^{\ell}\, w_i = \log_g x_i\right] > 1 - \mathsf{negl}(n).$$

The combiner has the following form:

$$\mathsf{H}^{(x_1, x_2, \cdots, x_\ell)}(r, m_0, m_1) = h^x(m_0, m_1) = g^{m_0} x^{m_1}.$$

where $x \leftarrow \mathsf{Z}(x_1, x_2, \cdots, x_\ell)$, and $r$ is the same random tape as the compression algorithm $\mathsf{Z}$ used.

Note that a pair of collisions for $h^y$ will give the discrete logarithm of $x$, which in turn can be used (by applying $\mathsf{U}$) to solve all DL instances $x_1, \ldots, x_\ell$, and therefore we can find a pair of collisions for each hash function $h^{x_i}$ efficiently. Thus this combiner is a $(1, \ell)$-combiner for CRHFs based on DL problem as defined in Definition 7. □

**Application of Special Hash Combiner: How to Set up a Global Hash.** Suppose in a multi-party setting, a given number of participants, $P_1, \cdots, P_\ell$, each $P_i$ has its own hash function $h^{x_i}$ with the same common parameter $\mathbb{G}, g$, and want to set up a *single* hash function trusted by all of them. The need for a global hash function was also addressed in [5]. While we can't simple choose some participant's hash function as the global hash function for obvious reasons. we can use our special hash combiner to solve this puzzle: Each participant runs the instance compression algorithm $\mathsf{Z}$ on these $(x_1, \cdots, x_l)$ locally and generates a single common $x \in \mathbb{G}$, and then they set $H_{(g,x)} : (m_0, m_1) \to g^{m_0} x^{m_1}$ to be the global hash function. This function is collision-resistant free since every collision would lead to a solution to the instance $x'$, which will enable the unfolding algorithm $\mathsf{U}$ to find all discrete logarithms of these random $x_i$'s, and thus if there is one $x_i$ generated at random by an honest party, no PPT algorithm can find a collision for $H_{(g,x)}$.

[9] Fully black box combiners mean both constructions and security proofs are black-box.

## 6 Open Problems

Our results also leave several interesting problems. The first one is to pinpoint the necessary and sufficient conditions on the hard distribution that admits constant-round public-coin witness hiding protocol. It is known that instance compression scheme is impossible with respect to NP-complete languages, and that the DL and RSA problems are unlikely to be NP-complete. We wonder if tailored instance compression schemes (with moderate parameters) exist for DL/RSA. It is shown that both positive and negative answers to this problem will have interesting consequences.

**Acknowledgments.** All authors are supported by the National Natural Science Foundation of China (Grant No. 61379141, No. 61772521), Key Research Program of Frontier Sciences, CAS (Grant No. QYZDB-SSW-SYS035), and the Open Project Program of the State Key Laboratory of Cryptology. The fourth author is also supported in by the National Key Research and Development Plan (Grant No. 2016YFB0800403), the National Natural Science Foundation of China (Grant No. 61772522) and Youth Innovation Promotion Association CAS.

## References

1. Bellare, M., Namprempre, C., Pointcheval, D., Semanko, M.: The one-more-RSA-inversion problems and the security of Chaum's blind signature scheme. J. Cryptol. **16**(3), 185–215 (2003)
2. Bellare, M., Palacio, A.: GQ and schnorr identification schemes: proofs of security against impersonation under active and concurrent attacks. In: Yung, M. (ed.) CRYPTO 2002. LNCS, vol. 2442, pp. 162–177. Springer, Heidelberg (2002). https://doi.org/10.1007/3-540-45708-9_11
3. Blum, M.: How to prove a theorem so no one else can claim it. In: ICM, pp. 1444–1451 (1986)
4. Brassard, G., Chaum, D., Crépeau, C.: Minimum disclosure proofs of knowledge. J. Comput. Syst. Sci. **37**(2), 156–189 (1988)
5. Canetti, R., Lin, H., Paneth, O.: Public-coin concurrent zero-knowledge in the global hash model. In: Sahai, A. (ed.) TCC 2013. LNCS, vol. 7785, pp. 80–99. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-36594-2_5
6. Canetti, R., Rivest, R., Sudan, M., Trevisan, L., Vadhan, S., Wee, H.: Amplifying collision resistance: a complexity-theoretic treatment. In: Menezes, A. (ed.) CRYPTO 2007. LNCS, vol. 4622, pp. 264–283. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-74143-5_15
7. Catalano, D., Fiore, D., Gennaro, R., Vamvourellis, K.: Algebraic (trapdoor) one-way functions and their applications. In: Sahai, A. (ed.) TCC 2013. LNCS, vol. 7785, pp. 680–699. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-36594-2_38
8. Cramer, R., Damgård, I., Schoenmakers, B.: Proofs of partial knowledge and simplified design of witness hiding protocols. In: Desmedt, Y.G. (ed.) CRYPTO 1994. LNCS, vol. 839, pp. 174–187. Springer, Heidelberg (1994). https://doi.org/10.1007/3-540-48658-5_19

9. Damgård, I., Nielsen, J.B.: Perfect hiding and perfect binding universally composable commitment schemes with constant expansion factor. In: Yung, M. (ed.) CRYPTO 2002. LNCS, vol. 2442, pp. 581–596. Springer, Heidelberg (2002). https://doi.org/10.1007/3-540-45708-9_37
10. Drucker, A.: New limits to classical and quantum instance compression. SIAM J. Comput. **44**(5), 1443–1479 (2015)
11. Dwork, C., Naor, M.: Zaps and their applications. In: FOCS, pp. 283–293. IEEE (2000)
12. Feige, U., Shamir, A.: Witness indistinguishable and witness hiding protocols. In: STOC, pp. 416–426 (1990)
13. Goldreich, O.: The Foundations of Cryptography. Basic Techniques, vol. 1. Cambridge University Press, Cambridge (2001)
14. Goldreich, O., Micali, S., Wigderson, A.: Proofs that yield nothing but their validity or all languages in NP have zero-knowledge proof systems. J. ACM **38**(3), 690–728 (1991)
15. Goldwasser, S., Micali, S., Rackoff, C.: The knowledge complexity of interactive proof systems. SIAM J. Comput. **18**(1), 186–208 (1989)
16. Groth, J., Ostrovsky, R., Sahai, A.: New techniques for noninteractive zero-knowledge. J. ACM **59**(3), 11:1–11:35 (2012)
17. Guillou, L.C., Quisquater, J.-J.: A practical zero-knowledge protocol fitted to security microprocessor minimizing both transmission and memory. In: Barstow, D., et al. (eds.) EUROCRYPT 1988. LNCS, vol. 330, pp. 123–128. Springer, Heidelberg (1988). https://doi.org/10.1007/3-540-45961-8_11
18. Haitner, I., Rosen, A., Shaltiel, R.: On the (im)possibility of arthur-merlin witness hiding protocols. In: Reingold, O. (ed.) TCC 2009. LNCS, vol. 5444, pp. 220–237. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-00457-5_14
19. Harnik, D., Naor, M.: On the compressibility of NP instances and cryptographic applications. SIAM J. Comput. **39**(5), 1667–1713 (2010)
20. Jain, A., Kalai, Y.T., Khurana, D., Rothblum, R.: Distinguisher-dependent simulation in two rounds and its applications. In: Katz, J., Shacham, H. (eds.) CRYPTO 2017. LNCS, vol. 10402, pp. 158–189. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-63715-0_6
21. Katz, J., Lindell, Y.: Introduction to Modern Cryptography. Chapman and Hall/CRC Press, London/Boca Raton (2007)
22. Maurer, U.: Zero-knowledge proofs of knowledge for group homomorphisms. Des. Codes Crypt. **77**(2–3), 663–676 (2015)
23. Pass, R.: Parallel repetition of zero-knowledge proofs and the possibility of basing cryptography on NP-hardness. In: IEEE CCC, pp. 96–110. IEEE (2006)
24. Pass, R.: Limits of provable security from standard assumptions. In: STOC, pp. 109–118. ACM (2011)
25. Peikert, C., Waters, B.: Lossy trapdoor functions and their applications. In: STOC, pp. 187–196. ACM (2008)
26. Pietrzak, K.: Non-trivial black-box combiners for collision-resistant hash-functions don't exist. In: Naor, M. (ed.) EUROCRYPT 2007. LNCS, vol. 4515, pp. 23–33. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-72540-4_2
27. Pietrzak, K.: Compression from collisions, or why CRHF combiners have a long output. In: Wagner, D. (ed.) CRYPTO 2008. LNCS, vol. 5157, pp. 413–432. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-85174-5_23
28. Schnorr, C.P.: Efficient identification and signatures for smart cards. In: Quisquater, J.-J., Vandewalle, J. (eds.) EUROCRYPT 1989. LNCS, vol. 434, pp. 688–689. Springer, Heidelberg (1990). https://doi.org/10.1007/3-540-46885-4_68

29. Shoup, V.: Lower bounds for discrete logarithms and related problems. In: Fumy, W. (ed.) EUROCRYPT 1997. LNCS, vol. 1233, pp. 256–266. Springer, Heidelberg (1997). https://doi.org/10.1007/3-540-69053-0_18
30. Zhang, J., Zhang, Z., Chen, Y., Guo, Y., Zhang, Z.: Black-box separations for one-more (static) CDH and its generalization. In: Sarkar, P., Iwata, T. (eds.) ASIACRYPT 2014. LNCS, vol. 8874, pp. 366–385. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-662-45608-8_20