On the Existence of 3-Round Zero-Knowledge Protocols *

Satoshi Hada †   Toshiaki Tanaka ‡

March 31, 1999

Abstract

In this paper, we construct a 3-round zero-knowledge protocol for any NP language. Our protocol achieves weaker notions of zero-knowledge than black-box simulation zero-knowledge. Therefore, our result does not contradict the triviality result of Goldreich and Krawczyk [GoKr96] which shows that 3-round black-box simulation zero-knowledge exist only for BPP languages. Our main contribution is to provide a non-black-box simulation technique. Whether there exists such a simulation technique was a major open problem in the theory of zero-knowledge. Our simulation technique is based on a non-standard computational assumption related to the Diffie-Hellman problem, which was originally proposed by Damgård [Da91]. This assumption, which we call the DA-1, says that, given randomly chosen instance of the discrete logarithm problem (p, q, g, g^a), it is infeasible to compute (B, X) such that $X = B^a \bmod p$ *without knowing* the value b satisfying $B = g^b \bmod p$.

Our protocol achieves different notions of zero-knowledge under different variants of DA-1. All variants are augmented by an auxiliary-input. Under non-uniform DA-1 w.r.t. auxiliary-input of *restricted* length, it achieves the notion of non-uniform zero-knolwedge. Furthermore, under the stronger variant called non-uniform DA-1 w.r.t. auxiliary-input of *non-restricted* length, it achieves the notion of auxiliary-input non-uniform zero-knowledge. Unfortunately, these variants have some kind of reverse-engineering property which seems to be unreasonable. It also achieves the standard notions called GMR zero-knowledge and auxiliary-input zero-knowledge under uniform variants of these assumptions, but they have the stronger reverse-engineering property.

Since the required computational assumptions seem to be unreasonable, our protocol may have no practical significance. However, our result is interesting in the sense that its zero-knowledge property can be demonstrated using the non-black-box simulation technique unlike almost all known zero-knowledge protocols. We note that it is still open whether one can construct a 3-round zero-knowledge protocol for a non-trivial language based on reasonable computational assumptions.

Keywords: Zero-knowledge, interactive proofs, interactive arguments, Diffie-Hellman problem, NP.

---

*An earlier version of this paper appeared in CRYPTO'98 [HT98]. We reformulated our main claim since our original one was wrong (See Appendix B).

†KDD R&D Laboratories, 2-1-15 Ohara, Kamifukuoka, Saitama 356-8502, Japan. `mailto:hada@lab.kdd.co.jp`.

‡KDD, 2-3-2 Nishishinjuku, Shinjuku-ku, Tokyo 163-8003, Japan. `mailto:tl-tanaka@kdd.co.jp`.

Contents

1 Introduction **3**
- 1.1 Background on Zero-Knowledge Protocol . . . 3
- 1.2 Classification of Zero-Knowledge . . . 3
- 1.3 Motivation and Contributions . . . 5
- 1.4 Our Results . . . 5
- 1.5 Organization . . . 6

2 Preliminaries **6**
- 2.1 Interactive Arguments . . . 6
- 2.2 Zero-Knowledge . . . 7
- 2.3 DLA and DHA . . . 9

3 Protocol 3R-ZK **9**
- 3.1 The Starting Point . . . 9
- 3.2 The Preliminary Transformation . . . 10
- 3.3 The Main Transformation . . . 11

4 Damgård's Assumption and Its Variants **13**
- 4.1 DA-1 . . . 13
- 4.2 Augmenting DA-1 by Auxiliary-Input . . . 14
- 4.3 DA-2 . . . 15

5 Correctness of 3R-ZK **15**
- 5.1 3R-ZK is an Argument . . . 16
- 5.2 3R-ZK is Zero-Knowledge . . . 18

6 Discussion **21**
- 6.1 Reverse-Engineering Property . . . 21
- 6.2 $Cl(AIZK)$ v.s. $Cl(BSZK)$ . . . 22
- 6.3 Statistical Zero-Knowledgeness . . . 22

7 Concluding Remarks **22**

Acknowledgments **23**

References **23**

A Blum's Zero-Knowledge Protocol for the Hamiltonian Circuit Problem **25**

B A Flaw in CRYPTO'98 version [HT98] **25**

1 Introduction

The fundamental notion of zero-knowledge (ZK) introduced by Goldwasser, Micali and Rackoff plays a central role in modern cryptography [GMR85]. In this paper, we investigate the methodology underlying ZK and construct a 3-round ZK protocol for NP.

1.1 Background on Zero-Knowledge Protocol

Consider an interactive protocol in which a prover convinces a verifier that a common input x belongs to some underlying language L (In this paper, L is in NP). The length of x is denoted by n and one measures complexity in terms of n. The verifier is always a probabilistic polynomial-time machine. We focus on two properties: "soundness" and "zero-knowledge." Each can be formalized in two ways depending on whether or not we restrict the adversary (the cheating prover and the cheating verifier) to a resource bounded machine.

The soundness asks that if $x \notin L$, any cheating prover can not convince the verifier to accept, except with negligible error probability. This notion is formalized in two ways: "proofs" and "arguments." These provide the statistical soundness and the computational soundness, respectively. The former requires that even a computationally unrestricted cheating prover should be unable to make the verifier accept $x \notin L$, except with negligible probability [GMR85]. On the other hand, the latter requires that any cheating prover restricted to probabilistic polynomial-time should be unable to make the verifier accept $x \notin L$, except with negligible probability [BrCr86][BCC88]. Although the notion of arguments is weaker than the notion of proofs, it is good enough for cryptographic applications. The soundness of arguments will typically depend on the complexity assumptions such as the discrete logarithm assumption. In this paper whenever we talk of proofs or arguments, we always mean ones with negligible error probability.

Zero-knowledge asks that when $x \in L$, an interaction with the prover yields no information (other than the fact $x \in L$) to any cheating verifier. This notion is also formalized in two ways: "statistical ZK" (SZK) and "computational ZK" (CZK). The former requires that even a computationally unrestricted cheating verifier will gain no useful information, except with negligible probability. On the other hand, the latter requires that any cheating verifier restricted to probabilistic polynomial-time will gain no useful information, except with negligible probability. Clearly, SZK is a special case of CZK.

In this paper, unless stated explicitly, ZK protocols mean CZK arguments. Our proposed protocol is a CZK argument.

1.2 Classification of Zero-Knowledge

There are several notions of ZK. Our proposed protocol achieves four different notions of ZK, called GMR zero-knowledge, non-uniform zero-knowledge, non-uniform auxiliary-input zero-knowledge, and auxiliary-input zero-knowledge, under different computational assumptions. We will begin by discussing the relationships among them.

ZK was originally formalized in [GMR85] as follows: For any probabilistic polynomial-time machine $\widehat{V}$ (the cheating verifier), there exists a probabilistic polynomial-time machine $S_{\widehat{V}}$ (the simulator) which produces a probability distribution which is computationally indistinguishable from the distribution of conversations of $\widehat{V}$ with the prover P. This original definition which we call GMR-ZK is not suitable for cryptographic applications since it is not closed under sequential composition [GoKr96]. In cryptographic applications, the cheating verifier may have some additional a-priori information.

In order to overcome the above problem, the notion of auxiliary-input zero-knowledge (AIZK) was introduced in [GoOr94] (The similar notion appeared in [TW87][GMR89]). AIZK is defined by augmenting GMR-ZK by an auxiliary-input. That is, the simulation requirement is extended to deal with non-uniform verifiers, which are allowed to take an auxiliary-input (an arbitrary string of polynomial length). The simulator is also allowed to take the same auxiliary-input used by the verifier. It was shown that AIZK is closed under sequential composition [GoOr94].

The above two notions only require that, for each cheating verifier $\widehat{V}$, there exists a simulator $S_{\widehat{V}}$. It is important to note that $S_{\widehat{V}}$ is allowed to examine the internal state of the cheating verifier $\widehat{V}$. On the other hand, black-box simulation zero-knowledge (BSZK) requires that there exists a single universal simulator that, using any non-uniform cheating verifier $\widehat{V}$ as a black-box, succeeds in simulating the interaction of $\widehat{V}$ with the prover P (Note that the order of quantifiers is reversed). That is, the simulator is only allowed to simply observe the input/output behavior of $\widehat{V}$. Although BSZK is most restrictive among three notions, almost all known ZK protocols are BSZK [1]. It was shown that BSZK implies AIZK [GoOr94].

All the above definitions are "semi-uniform" in the sense that it uses uniform machines but quantifies over all common inputs $x \in L$. In the totally non-uniform formalization of ZK, all machines are modeled by a polynomial-size circuit family [Go93]. We consider two non-uniform formalizations, i.e., non-uniform zero-knowledge and auxiliary-input non-uniform zero-knowledge.

Non-uniform zero-knowledge (NUZK) is a non-uniform variant of GMR-ZK. That is, it requires that for any polynomial-size circuit family $\widehat{V}$, there exists a (probabilistic) polynomial-size circuit family $S_{\widehat{V}}$ which produces a probability distribution which is computationally indistinguishable from the distribution of conversations of $\widehat{V}$ with the prover P. It is important to note that NUZK does not imply GMR-ZK [Go98b]. In fact, one can devise artificial protocols for sparse languages such that it achieves the notion of NUZK but not GMR-ZK. For example, consider the following interactive proof for a sparse language $L_{sp} = \{1^n\}_{n \in \mathsf{N}}$. The prover sends the verifier the value of a hard function $K(\cdot)$ on the common input $x \in L_{sp}$ (e.g., K is a non-recursive function indicating whether the nth Turing machine accepts every input). The verifier accepts if and only if x is of the form 1^n. Certainly, this is an interactive proof for L_{sp}. It is not GMR-ZK since there is no way to simulate in probabilistic polynomial-time the conversation in which the prover sends the value of $K(x)$. On the other hand, it is still NUZK since the simulator may just incorporate the hard bit $K(x)$. This shows that NUZK is a very weak notion of ZK and does not satisfy the intuitive requirement of ZK. Also, the result of [GoKr96] can be extended to show that NUZK is not closed under sequential composition.

Auxiliary-input non-uniform zero-knowledge (AINUZK) is defined by augmenting the notion of NUZK by an auxiliary-input. The above interactive proof for a sparse language achieves not only NUZK but also AINUZK. That is, AINUZK also does not satisfy the intuitive requirement of ZK. However, AINUZK has an advantage over NUZK since it is closed under sequential composition [GoOr94][Go93].

Let $Cl(def)$ denote the class of all interactive proofs and arguments satisfying the requirements of definition def. In light of the above, it holds that

$$Cl(BSZK) \subseteq Cl(AIZK) \subset Cl(GMR\text{--}ZK) \subset Cl(NUZK)$$

[1] The only exception is the contrived protocol constructed by Goldreich and Krawczyk for the purpose of separating GMR-ZK and AIZK [GoKr96]. They used the black-box simulation of uniform cheating verifier rather than non-uniform cheating verifier. However, their simulation technique does not seem useful for constructing 3-round zero-knowledge protocols for a non-trivial language.

and

$$Cl(AIZK) \subset Cl(AINUZK) \subset Cl(NUZK).$$

It is an open problem whether $Cl(BSZK)$ equals $Cl(AIZK)$ [GoOr94].

1.3 Motivation and Contributions

The round complexity, the number of messages exchanged, is a standard complexity measure for the efficiency of ZK protocols. Several researchers constructed constant round ZK protocols for NP [BCY89][FeSh89] [GoKa96] [2] [BJY97]. In particular, it is known that 4-round BSZK arguments exist for any NP language assuming the existence of one-way functions [BJY97].

The lower bounds on the round complexity have been investigated from the practical and theoretical viewpoint. Goldreich and Oren proved that 1-round GMR-ZK or 2-round AIZK protocols exist only for BPP languages [GoOr94]. Their result can be easily extended to show that 1-round NUZK or 2-round AINUZK protocols exist only for P/poly languages. Furthermore, Goldreich and Krawczyk proved that only languages in BPP have 3-round BSZK protocols [GoKr96] [3].

As mentioned above, almost all known ZK protocols are BSZK, where the simulator is only allowed to simply observe the input/output behavior of the cheating verifier. Also, there has been no known way to make use of the verifier's internal state so far. Therefore, many researchers considered that 4 is the minimal number of rounds required to satisfy the ZK property. However, since the argument in [GoKr96] uses the notion of black-box simulation in an essential way, their result does not apply to the weaker notions: NUZK, GMR-ZK, AINUZK and AIZK. That is, with respect to them, it is an interesting open problem whether there exists a 3-round ZK protocols for a non-trivial language. In other words, it asks whether there exists a non-black-box simulation technique which enables the simulator to make use of the internal state of any cheating verifier.

The main contribution of this paper is to provide such a non-black-box simulation technique. It is based on a non-standard computational assumption introduced by Damgård [Da91] [4]. We call it the DA-1. Loosely speaking, DA-1 says that, given randomly chosen instance of the discrete logarithm problem (p, q, g, g^a), it is infeasible to compute (B, X) such that $X = B^a \bmod p$ *without knowing* the value b satisfying $B = g^b \bmod p$. Damgård used DA-1 to construct a practical public-key cryptosystem provably secure against chosen-ciphertext attack. Interestingly, he pointed out that DA-1 implied a non-black-box simulation technique. However, he did not present a 3-round ZK protocol for a non-trivial language.

1.4 Our Results

We construct a 3-round ZK protocol for any NP language by transforming a 3-round honest-verifier ZK protocol into another 3-round protocol satisfying the ZK property against any cheating verifier. Our transformation preserves the statistical zero-knowledgeness. That is, if the starting protocol is SZK, the resultant protocol is SZK as well. However, the perfect zero-knowledgeness can not be preserved.

Our proposed protocol is an interactive argument under DA-2 which is a variant of DA-1. We will show that, using our non-black-box simulation technique, it achieves different notions of ZK (NUZK, GMR-ZK, AINUZK, AIZK) under different variants of DA-1. Table 1 summarizes the

[2]The ZK protocol constructed in [GoKa96] is a proof rather than an argument.

[3]The discussion in [GoKr96] are for CZK proofs. However, their result extends to CZK arguments. See Remarks 6.3 and 6.5 in that paper.

[4][HT98] missed this reference since we were unaware of it.

achieved ZK notions. Unfortunately, all results are problematic since all variants of DA-1 have some kind of reverse-engineering property which seems to be unreasonable. In particular, the case of AIZK seems to be most problematic since uniform DA-1 w.r.t. auxiliary-input of non-restricted length has the strongest reverse-engineering property. We don't know whether one can get rid of the reverse-engineering property. It is still open whether one can construct a 3-round zero-knowledge protocol for a non-trivial language based on reasonable computational assumptions.

Table 1: Summary of achieved ZK notions

Assumption	Achieved notion	Comment
non-uniform DA-1 w.r.t. aux-input of restricted length	NUZK	less problematic
non-uniform DA-1 w.r.t. aux-input of non-restricted length	AINUZK	
uniform DA-1 w.r.t. aux-input of restricted length	GMR-ZK	
uniform DA-1 w.r.t. aux-input of non-restricted length	AIZK	most problematic

1.5 Organization

In Section 2, we give the definitions of interactive arguments, several ZK notions, and some standard complexity assumptions. Section 3 describes our proposed protocol. In Section 4, we give Damgård's assumption and its variants. In Section 5, we prove the correctness of our protocol. In Section 6, we discuss the properties and the implications of our results. In Section 7, we conclude with some remarks.

2 Preliminaries

In this section, we give the definitions of ZK arguments and some standard complexity assumptions.

Our notations for probabilistic machines, spaces and experiments follow [BeRo93]. If A is a probabilistic machine then $A(x_1, x_2, \cdots)$ denotes the output distribution of A on inputs $(x_1, x_2, \cdots)$. If S is any probability distribution then $x \leftarrow S$ denotes the operation of selecting an element according to S. If S is a set then we use the same notation to denote the operation of picking an element x uniformly from S. For deterministic machines C (such as circuits), we write $y = C(x_1, x_2, \cdots)$ to mean that the output of C on inputs $(x_1, x_2, \cdots)$ is y. $\Pr[x_1 \leftarrow S_1; x_2 \leftarrow S_2; \cdots : p(x_1, x_2, \cdots)]$ denotes the probability that the predicate $p(x_1, x_2, \cdots)$ is true after the ordered execution of the operation $x_1 \leftarrow S_1, x_2 \leftarrow S_2, \cdots$.

We deal with NP languages and let $W_L(x)$ denote the witness set of x which belongs to an NP language L. We say that a function $\nu(\cdot) : \mathsf{N} \rightarrow \mathsf{R}$ is negligible if for every polynomial $poly(\cdot)$ and all sufficiently large n's, it holds that $\nu(n) < 1/poly(n)$. Also, we say that a function $g(\cdot) : \mathsf{N} \rightarrow \mathsf{R}$ is overwhelming if $g(\cdot) = 1 - \nu(\cdot)$ for some negligible function $\nu(\cdot)$.

2.1 Interactive Arguments

We consider two probabilistic polynomial-time interactive machines called the prover and the verifier. Initially both machines have access to a common input tape which includes x of length n. The prover and the verifier send messages to one another through two communication tapes. After exchanging a polynomial number of messages, the verifier stops in an accept state or in a reject state. Each machine, denoted by A, only sees its own tapes, namely, the common input tape, the random tape, the auxiliary-input tape and the communications tapes. Typically, the prover's auxiliary-input tape includes a witness $w \in W_L(x)$. Let $A(x, y, m; R)$ denote A's next message when x is the common input, y the auxiliary-input, R the random coins and m the messages exchanged so far. We omit R when it is unnecessary to make the random coins explicit. That is, $A(x, y, m)$ denotes the distribution of A's next message.

The view of the verifier $\widehat{V}$, denoted by $[x, y, m; R]$, consists of the common input x, $\widehat{V}$'s auxiliary-input y, $\widehat{V}$'s random coins R and the sequence m of messages exchanged by the prover and the verifier during the interaction. Let $\langle P(x, w), \widehat{V}(x, y)\rangle$ denote the probability distribution of $\widehat{V}$'s view when interacting with P on the common input x, where w is the auxiliary-input to P, and y the auxiliary-input to $\widehat{V}$. Let $\rho_V(x, m, R)$ denote the deterministic computation used by the prescribed verifier V to determine whether to accept or to reject. Let $\mathsf{Acc}\langle P(x, w), V(x, y)\rangle$ denote the probability that $\langle P(x, w), V(x, y)\rangle$ is accepting, that is,

$$\mathsf{Acc}\langle P(x, w), V(x, y)\rangle = \Pr\left[\ [x, y, m; R] \leftarrow \langle P(x, w), V(x, y)\rangle; \rho_V(x, m, R) = \mathsf{Accept}\ \right].$$

Definition 2.1 [interactive arguments [Go98a]] Let P, V be two probabilistic polynomial-time interactive machines. We say that (P, V) is an interactive argument for L if the following two conditions are satisfied:

- Efficient Completeness: For every $x \in L$, there exists a string w, $\mathsf{Acc}\langle P(x, w), V(x, -)\rangle = 1$.
- Computational Soundness: For every probabilistic polynomial-time machine $\widehat{P}$ (the cheating prover), every polynomial $poly(\cdot)$, all sufficiently long $x \notin L$ and all w's,

$$\mathsf{Acc}\langle \widehat{P}(x, w), V(x, -)\rangle < \frac{1}{poly(|x|)}.$$

In both conditions, the auxiliary-input to the verifier V is empty. The auxiliary-input w to P is typically a witness of the common input x, i.e., $w \in W_L(x)$. We note that the cheating prover $\widehat{P}$ is allowed to take an auxiliary-input. This simplifies the proof of the soundness of our proposed protocol in Section 5.

2.2 Zero-Knowledge

Here, we give the definitions of GMR-ZK, AIZK, NUZK and AINUZK. We first give a general definition and then define GMR-ZK and AIZK as special cases. We give the definitions of NUZK and AINUZK in the same way. Let $P_L(x)$ denote the set of strings w satisfying the efficient completeness condition with respect to $x \in L$. We basically follow the definition given in [Go98a].

Definition 2.2 [zero-knowledge w.r.t. auxiliary-input of length l_{ZK_AI}] Let $l_{ZK\_AI}(\cdot)$ be a polynomial. Let P, V be two probabilistic polynomial-time interactive machines. We say that (P, V) is zero-knowledge w.r.t. auxiliary-input of length l_{ZK_AI} for L if for every probabilistic polynomial-time machine $\widehat{V}$ (the cheating verifier), there exists a probabilistic polynomial-time machine $S_{\widehat{V}}$ (the simulator) such that the following two ensembles are computationally indistinguishable:

$$\{S_{\widehat{V}}(x, y)\}_{x \in L, y \in \{0,1\}^{l_{ZK\_AI}(|x|)}} \text{ and } \{\langle P(x, w), \widehat{V}(x, y)\rangle\}_{x \in L, w \in P_L(x), y \in \{0,1\}^{l_{ZK\_AI}(|x|)}}.$$

Namely, for every polynomial-size circuit family $D = \{D_{x,y}\}_{x \in L, y \in \{0,1\}^{l_{ZK\_AI}(|x|)}}$, every polynomial $poly(\cdot)$, all sufficiently long $x \in L$ and all $y \in \{0,1\}^{l_{ZK\_AI}(|x|)}$,

$$|\Pr\left[v \leftarrow S_{\widehat{V}}(x, y) : D_{x,y}(v) = 1\right] - \Pr\left[v \leftarrow \langle P(x, w), \widehat{V}(x, y)\rangle : D_{x,y}(v) = 1\right]| < \frac{1}{poly(|x|)}.$$

If the above two ensembles are statistically close (resp., identical), we say that (P, V) is statistical (resp., perfect) zero-knowledge w.r.t. auxiliary-input of length l_{ZK_AI} for L

We give the definitions of GMR-ZK and AIZK as special cases.

Definition 2.3 [GMR zero-knowledge] We say that (P, V) is GMR zero-knowledge for L if it is zero-knowledge w.r.t. auxiliary-input of length $l_{ZK\_AI} = 0$ for L. We say that (P, V) is GMR statistical (resp., perfect) zero-knowledge for L if it is statistical (resp., perfect) zero-knowledge w.r.t. auxiliary-input of length $l_{ZK\_AI} = 0$ for L.

Definition 2.4 [auxiliary-input zero-knowledge] We say that (P, V) is auxiliary-input zero-knowledge for L if for every polynomial l_{ZK_AI}, it is zero-knowledge w.r.t. auxiliary-input of length l_{ZK_AI} for L. We say that (P, V) is auxiliary-input statistical (resp., perfect) zero-knowledge for L if for every polynomial l_{ZK_AI}, it is statistical (resp., perfect) zero-knowledge w.r.t. auxiliary-input of length l_{ZK_AI} for L.

Next, we define NUZK and AINUZK in the same way. They are non-uniform variants of GMR-ZK and AIZK.

Definition 2.5 [non-uniform zero-knowledge w.r.t. auxiliary-input of length l_{ZK_AI}] Let $l_{ZK\_AI}(\cdot)$ be a polynomial. Let P, V be two probabilistic polynomial-time interactive machines. We say that (P, V) is non-uniform zero-knowledge w.r.t. auxiliary-input of length l_{ZK_AI} for L if for every polynomial-size circuit family $\widehat{V}$ (the cheating verifier), there exists a *probabilistic* polynomial-size circuit family $S_{\widehat{V}}$ (the simulator) such that the following two ensembles are computationally indistinguishable:

$$\{S_{\widehat{V}}(x, y)\}_{x \in L, y \in \{0,1\}^{l_{ZK\_AI}(|x|)}} \text{ and } \{\langle P(x, w), \widehat{V}(x, y)\rangle\}_{x \in L, w \in P_L(x), y \in \{0,1\}^{l_{ZK\_AI}(|x|)}}.$$

If the above two ensembles are statistically close (resp., identical), we say that (P, V) is non-uniform statistical (resp., perfect) zero-knowledge w.r.t. auxiliary-input of length l_{ZK_AI} for L

We note that the simulator (polynomial-size circuit family) is allowed to toss the random coins. That is, the simulator takes an auxiliary-input which is uniformly selected. Otherwise, two ensembles are always distinguishable since the output of the simulator is deterministic.

Definition 2.6 [non-uniform zero-knowledge] We say that (P, V) is non-uniform zero-knowledge for L if it is non-uniform zero-knowledge w.r.t. auxiliary-input of length $l_{ZK\_AI} = 0$ for L. We say that (P, V) is non-uniform statistical (resp., perfect) zero-knowledge for L if it is non-uniform statistical (resp., perfect) zero-knowledge w.r.t. auxiliary-input of length $l_{ZK\_AI} = 0$ for L.

Definition 2.7 [auxiliary-input non-uniform zero-knowledge] We say that (P, V) is auxiliary-input non-uniform zero-knowledge for L if for every polynomial $l_{ZK\_AI}(\cdot)$, it is non-uniform zero-knowledge w.r.t. auxiliary-input of length l_{ZK_AI} for L. We say that (P, V) is auxiliary-input non-uniform statistical (resp., perfect) zero-knowledge for L if for every polynomial $l_{ZK\_AI}(\cdot)$, it is non-uniform statistical (resp., perfect) zero-knowledge w.r.t. auxiliary-input of length l_{ZK_AI} for L.

We remark that in all definitions above, it is not required that, given a verifier $\widehat{V}$, the simulator $S_{\widehat{V}}$ can be effectively constructed, but rather that it exists.

2.3 DLA and DHA

We give two complexity assumptions related to the discrete logarithm problem. All exponentiations in this paper are in Z_p^*. The definition of the prime p will be clear by the context. To simplify the notations, we omit the expression "mod p".

First, we recall the discrete logarithm assumption (DLA). In this paper, we need a stronger formulation (than usual one) in which we assume nothing on the distribution of the prime p and the base g.

Definition 2.8 Let L_{PQG} denote the set $\{(p, q, g)\}$ of primes and generators, where p and q are primes such that $p = 2q + 1$ and g is an element of order q in Z_p^* (a generator of a subgroup of Z_p^* of order q).

L_{PQG} can be recognized in probabilistic polynomial-time with negligible error probability by testing primality for p and q in probabilistic polynomial-time [SS77][Ra80], and verifying that g is not the identity and that $g^q = 1$. Furthermore, there exists a probabilistic polynomial-time machine which, on input 1^n, outputs $(p, q, g) \in L_{PQG}$ such that p is of length n.

Assumption 2.9 [non-uniform DLA] For every family of polynomial-size circuits $I = \{I_n\}_{n\geq 1}$, every polynomial $poly(\cdot)$, all sufficiently large n's and all $(p, q, g) \in L_{PQG}$ such that p is of length n,

$$\Pr[a \leftarrow Z_q : I_n(p, q, g, g^a) = a] < \frac{1}{poly(n)}.$$

Next, we give the definition of the Diffie-Hellman assumption (DHA) which says that the Diffie-Hellman problem [DH76] is intractable in the same setting as the above definition of DLA.

Assumption 2.10 [non-uniform DHA] For every family of polynomial-size circuits $I = \{I_n\}_{n\geq 1}$, every polynomial $poly(\cdot)$, all sufficiently large n's and all $(p, q, g) \in L_{PQG}$ such that p is of length n,

$$\Pr[a \leftarrow Z_q; b \leftarrow Z_q : I_n(p, q, g, g^a, g^b) = g^{ab}] < \frac{1}{poly(n)}.$$

Although uniform DLA and uniform DHA are defined analogously, we omit them here.

3 Protocol 3R-ZK

In this section, we describe a 3-round ZK protocol (called the 3R-ZK) for any NP language. We construct it by combining two transformations.

3.1 The Starting Point

Our starting point is a 3-round honest-verifier ZK protocol (A, B) for an NP-complete language L. We call it the 3R-HVZK. Let M_1, Y, M_2 denote the messages exchanged in the protocol. M_1 and M_2 are the first and the second messages sent by the prover, respectively. Y is the message sent by the verifier. We require that the protocol 3R-HVZK satisfies the following properties.

B0. The message Y is chosen uniformly at random from a polynomially sampleable subdomain D_Y of $\{0, 1\}^n$, where n is the length of the common input x.

B1. The prover A can be implemented in probabilistic polynomial-time when it is given as its auxiliary-input an NP witness $w \in L_W(x)$.

B2. It satisfies a *strong soundness* property which requires that for every common input $x \notin L$ and every possible first message M_1, there exists at most one verifier's message Y such that the prover can answer properly in its second message M_2. This means that if the size of D_Y is superpolynomial in n then the probability that the verifier accepts $x \notin L$ is negligible.

B3. It satisfies the ZK property with respect to a prescribed verifier B, i.e., honest-verifier zero-knowledge (HVZK). That is, there exists a simulator which produces a probability distribution computationally indistinguishable from the distribution of the real view of B.

The typical example of 3R-HVZK is the parallel composition of Blum's ZK protocol for the Hamiltonian circuit problem [Bl86] (See Appendix A). The following theorem summarizes our starting point.

Theorem 3.1 [Starting point [Bl86]] Assuming the existence of non-uniformly secure commitment scheme, there exists a 3-round protocol (A, B) satisfying the properties B0, B1, B2 and B3 for any NP language.

3.2 The Preliminary Transformation

In the preliminary transformation, we just modify 3R-HVZK so that the message Y of the verifier are randomly chosen from a domain D_Y specified by the first message of the prover. We denote by $(\bar{P}, \bar{V})$ the resultant protocol.

Protocol: $(\bar{P}, \bar{V})$ for an NP-complete language L.

Common Input: a common input x of length n.

Prover's Aux-Input: an NP witness w in $W_L(x)$.

P1: The prover $\bar{P}$ generates an instance $(p, q, g) \in L_{PQG}$ such that p is of length n. $\bar{P}$ also computes M_1 according to (A, B). Then $\bar{P}$ sends (M_1, p, q, g) to the verifier $\bar{V}$.

V1: $\bar{V}$ checks whether (p, q, g) is in L_{PQG} and p is of length n. If this is true then $\bar{V}$ chooses Y uniformly at random in a subgroup of Z_p^* of order q and send Y to $\bar{P}$.

P2: $\bar{P}$ computes M_2 according to (A, B) and sends it to $\bar{V}$.

V2: $\bar{V}$ checks whether M_2 is valid according to (A, B). If this is true then $\bar{V}$ accepts, otherwise $\bar{V}$ rejects.

It is easy to see that the protocol $(\bar{P}, \bar{V})$ satisfies the properties B1, B2 and B3. B1 and B2 are clearly satisfied. B3 is also satisfied because of the properties B0 and B3 of 3R-HVZK. We restate the property B3 of $(\bar{P}, \bar{V})$ formally as follows:

B3. For the prescribed verifier $\bar{V}$, there exists a probabilistic polynomial-time simulator S_{HV} (the honest-verifier simulator) such that the following two ensembles are computationally indistinguishable:

$$\{S_{HV}(x)\}_{x \in L} \text{ and } \{\langle \bar{P}(x, w), \bar{V}(x, -)\rangle\}_{x \in L, w \in W_L(x)}.$$

Namely, for every polynomial-size circuit family $D = \{D_x\}_{x \in L}$, every polynomial $poly(\cdot)$, and all sufficiently long $x \in L$,

$$|\Pr[v \leftarrow S_{HV}(x) : D_x(v) = 1] - \Pr[v \leftarrow \langle \bar{P}(x, w), \bar{V}(x, -)\rangle : D_x(v) = 1]| < \frac{1}{poly(|x|)}.$$

We note that the output of $S_{HV}(x)$ is of the form $[x, -, (M_1, p, q, g, Y, M_2); R_Y]$, where the auxiliary-input to $\bar{V}$ is empty, R_Y denotes $\bar{V}$'s random coins used to compute Y and Y is an element of Z_p^* of order q.

We use this honest-verifier simulator S_{HV} to demonstrate the zero-knowledge property of 3R-ZK in Section 5,

3.3 The Main Transformation

Now we present our main transformation. Before describing it, we roughly review the approach used by Feige and Shamir to construct a 4-round ZK protocol for NP [FeSh89]. First of all, their protocol is a "secret-coin" protocol. That is, the messages sent by the verifier are not just random coins, but they are computed using secret random coins. Their protocol satisfies the following properties.

R1. The verifier proves to the prover that he knows the secret coins inducing the messages sent by him.

R2. A simulator is able to get the secret coins used by the cheating verifier no matter how he behaves. Furthermore, once the simulator get the secret coins, the simulation is easy to complete. As a result, the ZK property is satisfied.

R3. Any cheating prover can not get the secret coins. Furthermore, the protocol is designed so that the computational soundness is satisfied under this property.

In [FeSh89], these properties have been implemented using the notion of witness-hiding proofs of knowledge [FeSh90][BeGo92]. The verifier executes a witness-hiding proof of knowledge of the secret coins. The property R3 is satisfied by the witness-hiding property [FeSh90]. R1 and R2 are also satisfied by the notion of "proof of knowledge" [BeGo92]. That is, the simulator is able to extract the secret coins of the cheating verifier by using it as a black-box and "rewinding" it. The computational soundness condition is satisfied simultaneously since any cheating prover is not allowed to "rewind" the verifier.

We follow their approach to construct 3R-ZK, but the notion of witness hiding proofs does not seem useful for our purpose since proofs of knowledge require an interaction (i.e., the verifier must send at least two messages) which we can not afford here. Nevertheless, we overcome this problem using a non-standard computational assumption introduced by Damgård [Da91].

Now we transform $(\bar{P}, \bar{V})$ into our proposed protocol 3R-ZK (P, V) satisfying all properties R1, R2 and R3.

Protocol: 3R-ZK (P, V) for an NP-complete language L.

Common Input: a common input x of length n.

Prover's Aux-Input: an NP witness w in $W_L(x)$.

P1: The prover P computes the first message as follows:

P1-1: P generates (M_1, p, q, g) according to $(\bar{P}, \bar{V})$.

P1-2: P generates uniformly a random number $a \in Z_q$ and computes $A = g^a$.

P sends (M_1, p, q, g, A) to the verifier V.

V1: V checks whether (p, q, g) is in L_{PQG} and p is of length n. If this is true then V generates a random number $b \in Z_q$, computes $(B, X) = (g^b, A^b)$ and sends (B, X) to P. Otherwise V rejects. Note that b is the secret-coin used by V.

P2: P checks whether $X = B^a$. If this is false then P stops, otherwise P computes the second message as follows:

P2-1: P generates a random number $c \in Z_q$ and computes $(C, Y) = (g^c, B^c)$ (These may also be computed as $(C, Y) = (A^c, X^c)$).

P2-2: According to $(\bar{P}, \bar{V})$, P computes M_2 using Y as the message sent by $\bar{V}$.

P sends (M_2, C, Y) to V.

V2: V checks whether the following two conditions are satisfied:

V2-1: V checks whether $Y = C^b$.

V2-2: According to $(\bar{P}, \bar{V})$, V checks whether M_2 is valid using Y as the message sent by $\bar{V}$.

If either condition is violated then V rejects, otherwise V accepts.

We explain that 3R-ZK satisfies all properties R1, R2, and R3. We can make the following observation on the computation of V in V1: It seems that any cheating verifier must start by simply choosing b and compute (g^b, A^b) in order to pass the check $X = B^a$ by the prover in P2. That is, we assume that *the verifier knows the secret coins b whenever it holds that $X = B^a$*. This means that the property R1 is satisfied. Informally, this assumption can be defined as follows: For any cheating verifier $\widehat{V}$, there exists another verifier $\widehat{V}'$ who outputs not only (B, X) but also the secret coins b whenever $\widehat{V}$ outputs (B, X) satisfying $X = B^a$. Therefore, a simulator is able to get the secret coins b by using $\widehat{V}'$ as a black-box rather than $\widehat{V}$. Once the simulator gets b, the simulation is easy to do since given Y and b, it is easy to compute the message C such that $Y = C^b$. As a result, the property R2 is satisfied. We stress that the simulator does not use the cheating verifier $\widehat{V}$ as a black-box. The above assumption is very similar to the one proposed by Damgård in [Da91] which we call the DA-1. However, precisely speaking, it is different from DA-1 due to the common input x, the message M_1 and the auxiliary-input y to the cheating verifier. We must augment DA-1 by an auxiliary-input in order to demonstrate the ZK property.

The random coins b used by V are kept secret from any cheating prover if we assume that it is computationally intractable to compute b from (g, A, B, X). Now we focus on the computation of P in P2-1. As in the case of the computation of V in V1, it seems that any cheating prover must start by simply choosing c and compute either (g^c, B^c) or (A^c, X^c) in order to pass the check $Y = C^b$ by the verifier in V2-1 (Note that there are two ways of computing (C, Y) in P2-1 to pass it). We assume that *the prover knows the value of c such that $Y = B^c$ or X^c whenever it holds $Y = C^b$*. We call this assumption the DA-2, which is defined informally as follows: For any cheating prover $\widehat{P}$, there exists another prover $\widehat{P}'$ who outputs not only (C, Y) but also the secret coins c whenever $\widehat{P}$ outputs (C, Y) satisfying $Y = C^b$. Assume that the computational soundness is not satisfied. Then, given a fixed message Y, the cheating prover $\widehat{P}$ must compute the second message C satisfying $Y = C^b$ for randomly chosen (B, X). Under DA-2, this means that, given a fixed message Y, $\widehat{P}'$ can compute C and its discrete logarithm c such that $Y = B^c$ or X^c for randomly chosen (B, X). This contradicts DLA. Therefore, the property R3 is satisfied.

The above observations roughly show that 3R-ZK is a ZK argument for L. The formal proofs are given in Section 5. Before giving it, we review the formal definitions of DA-1 and give several augmentations of DA-1 in next section.

4 Damgård's Assumption and Its Variants

In this section, we review Damgård's assumption introduced in [Da91] and give its variants. As described in the previous section, there are two versions: DA-1 and DA-2. They are required for the ZK property and the computational soundness, respectively. First, we give the definitions of DA-1. Then we augment DA-1 by an auxiliary-input. Lastly, we give the definition of DA-2.

4.1 DA-1

DA-1 can be formulated both in the uniform model and in the non-uniform model as other computational assumptions. The original assumption in [Da91] is called uniform DA-1 formalized as follows:

Assumption 4.1 [uniform DA-1[Da91]] Let M be a probabilistic polynomial-time machine which takes as input (p, q, g, g^a) and tries to output (B, X) such that $X = B^a$. For every probabilistic polynomial-time machine M, there exists a probabilistic polynomial-time machine M' which, on input (p, q, g, g^a), outputs (B', X', b) satisfying the following two conditions.

1. Assume that M and M' use the same random coins. Then, for every $(p, q, g) \in L_{PQG}$, it holds that

$$\Pr[a \leftarrow Z_q; (B, X) \leftarrow M(p, q, g, g^a); (B', X', b) \leftarrow M'(p, q, g, g^a) : (B', X') = (B, X)] = 1.$$

2. For every polynomial $poly(\cdot)$, all sufficiently large n's and every $(p, q, g) \in L_{PQG}$ such that p is of length n,

$$\Pr[a \leftarrow Z_q; (B', X', b) \leftarrow M'(p, q, g, g^a) : X' = B'^a \ \wedge \ B' \neq g^b \ ] < \frac{1}{poly(n)}.$$

Roughly, the above conditions say that whenever M outputs (B, X) such that $X = B^a$, M' outputs not only (B, X) but also b such that $B = g^b$ (i.e., $X = (g^a)^b$) with overwhelming probability.

The non-uniform DA-1 is formalized as follows.

Assumption 4.2 [non-uniform DA-1] Let $M = \{M_n\}_{n \geq 1}$ be a polynomial-size circuit family which takes as input (p, q, g, g^a) and tries to output (B, X) such that $X = B^a$, where M_n runs on p of length n. For every polynomial-size circuit family $M = \{M_n\}_{n \geq 1}$, there exists a polynomial-size circuit family $M' = \{M'_n\}_{n \geq 1}$ which, on input (p, q, g, g^a), outputs (B', X', b) satisfying the following two conditions.

1. For every $(p, q, g) \in L_{PQG}$, it holds that

$$\Pr[a \leftarrow Z_q; (B, X) = M_n(p, q, g, g^a); (B', X', b) = M'_n(p, q, g, g^a) : (B', X') = (B, X)] = 1.$$

2. For every polynomial $poly(\cdot)$, all sufficiently large n's and every $(p, q, g) \in L_{PQG}$ such that p is of length n,

$$\Pr[a \leftarrow Z_q; (B', X', b) = M'_n(p, q, g, g^a) : X' = B'^a \ \wedge \ B' \neq g^b \ ] < \frac{1}{poly(n)}.$$

Clearly, uniform DA-1 implies non-uniform DA-1, but we don't know whether the reverse implication holds. The proposition below shows that the combination of DA-1 and DLA implies DHA. However, it is unlikely that the reverse implication holds.

Proposition 4.3 Under uniform DA-1 and uniform DLA, uniform DHA holds. Under non-uniform DA-1 and non-uniform DLA, non-uniform DHA holds.

Proof: We give the proof only in the uniform case. Assume that uniform DHA does not hold. Then there exists a probabilistic polynomial-time machine M which, on input $p, q, g, A = g^a, B = g^b$, outputs $B = g^b$ and $X = g^{ab}$ with not negligible probability. Applying uniform DA-1 to this machine M, we have another probabilistic polynomial-time machine M' which, on inputs $p, q, g, A = g^a, B = g^b$, outputs not only (B, X) but also the discrete logarithm b such that $B = g^b$. This contradicts uniform DLA. ▮

4.2 Augmenting DA-1 by Auxiliary-Input

We need several variants of DA-1 in order to prove that 3R-ZK satisfies the ZK property. They say that DA-1 holds even when M is allowed to take an auxiliary-input. We first give a general augmentation of DA-1 and then give two assumptions as special cases. One is called DA-1 w.r.t. auxiliary-input of *non-restricted* length. The other is called DA-1 w.r.t. auxiliary-input of *restricted* length. The former requires that DA-1 holds when M is allowed to take an auxiliary-input of any polynomial length. On the other hand, the latter requires that DA-1 holds when M is allowed to take an auxiliary-input of pre-specified length. Clearly, the former is stronger than the latter. We note that these variants have some kind of reverse-engineering property. This issue is taken up in Section 6. Here, we give the definitions only in the non-uniform model. Uniform definitions are defined analogously, but we omit them here.

Assumption 4.4 [non-uniform DA-1 w.r.t. auxiliary-input of length l_{DA_AI}] Let $l_{DA\_AI}(\cdot)$ be a polynomial. Let $M = \{M_n\}_{n\geq 1}$ be a polynomial-size circuit family which takes as input (p, q, g, g^a, aux) and tries to output (B, X) such that $X = B^a$, where M_n runs on p of length n. For every polynomial-size circuit family $M = \{M_n\}_{n\geq 1}$, there exists a polynomial-size circuit family $M' = \{M'_n\}_{n\geq 1}$ which, on input (p, q, g, g^a), outputs (B', X', b) satisfying the following two conditions.

1. For every $(p, q, g) \in L_{PQG}$ and every $aux \in \{0,1\}^{l_{DA\_AI}(n)}$, it holds that

$$\Pr\left[\begin{array}{l} a \leftarrow Z_q; (B, X) = M_n(p, q, g, g^a, aux); \\ (B', X', b) = M'_n(p, q, g, g^a, aux) \end{array} : (B', X') = (B, X)\right] = 1.$$

2. For every polynomial $poly(\cdot)$, all sufficiently large n's, every $(p, q, g) \in L_{PQG}$ such that p is of length n and every $aux \in \{0,1\}^{l_{DA\_AI}(n)}$,

$$\Pr[a \leftarrow Z_q; (B', X', b) = M'_n(p, q, g, g^a, aux) : X' = B'^a \ \wedge \ B' \neq g^b\ ] < \frac{1}{poly(n)}.$$

Assumption 4.5 [non-uniform DA-1 w.r.t. auxiliary-input of restricted length] For a pre-specified polynomial $l_{DA\_AI}(\cdot)$, non-uniform DA-1 w.r.t. auxiliary-input l_{DA_AI} holds.

Assumption 4.6 [non-uniform DA-1 w.r.t. auxiliary-input of non-restricted length] For every polynomial $l_{DA\_AI}(\cdot)$, non-uniform DA-1 w.r.t. auxiliary-input l_{DA_AI} holds.

These variants clearly imply non-uniform DA-1, but we don't know whether the reverse implications hold.

4.3 DA-2

We formulate DA-2 in the non-uniform model. Although uniform DA-2 is formulated similarly, we do not need it in this paper.

Assumption 4.7 [non-uniform DA-2] Let M be a polynomial-time circuit family which takes as input $(p, q, g, g^a, g^b, g^{ab})$ and tries to output (C, Y) such that $Y = C^b$, where M_n runs on p of length n. For every polynomial-size circuit family $M = \{M_n\}_{n\geq 1}$, there exists a polynomial-size circuit family $M' = \{M'_n\}_{n\geq 1}$ which takes as input $(p, q, g, g^a, g^b, g^{ab})$ and outputs (C', Y', c) satisfying the following two conditions.

1. For every $(p, q, g) \in L_{PQG}$ such that p is of length n and every $a \in Z_q$, it holds that

$$\Pr\left[\begin{array}{l} b \leftarrow Z_q; (C, Y) = M_n(p, q, g, g^a, g^b, g^{ab}); \\ (C', Y', c) = M'_n(p, q, g, g^a, g^b, g^{ab}) \end{array} : (C', Y') = (C, Y)\right] = 1.$$

2. For every polynomial $poly(\cdot)$, all sufficiently large n's, every $(p, q, g) \in L_{PQG}$ such that p is of length n and every $a \in Z_q$,

$$\Pr\left[\begin{array}{l} b \leftarrow Z_q; \\ (C', Y', c) = M'_n(p, q, g, g^a, g^b, g^{ab}) \end{array} : Y' = C'^b \ \wedge \ Y' \neq (g^b)^c \ \wedge \ Y' \neq (g^{ab})^c \right] < \frac{1}{poly(n)}.$$

Roughly, the above conditions say that whenever M outputs (C, Y) such that $Y = C^b$, M' outputs not only (C, Y) but also c such that $Y = (g^b)^c$ or $Y = (g^{ab})^c$ (i.e., $C = g^c$ or $(g^a)^c$) with overwhelming probability.

Although it is clear that DA-2 implies DA-1, we don't know whether the reverse implication holds.

In all assumptions above, it is not required that, given M, M' can be effectively constructed, but rather that it exists. This is reminiscent of the definitions of ZK given in Section 2.

Remark 4.8 DA-1 and DA-2 are fundamentally different from the standard complexity assumptions such as DLA and DHA in the sense that they have double quantification (i.e., for every adversary, there exists another one such that something holds), whereas the standard assumptions have only one quantifier (i.e., for every adversary, something holds). It is unlikely that they are proved to hold under standard complexity assumptions. We believe that it is interesting to study their validity.

5 Correctness of 3R-ZK

Theorem 5.1 summarizes the properties of our proposed protocol 3R-ZK.

Theorem 5.1 [Main Theorem] Assume that 3R-HVZK for L satisfies the properties B0, B1, B2 and B3. Then the protocol 3R-ZK for L is:

(0) an argument for L under non-uniform DA-2 and non-uniform DLA.

(1) NUZK for L under non-uniform DA-1 w.r.t auxiliary-input of restricted length.

(2) AINUZK for L under non-uniform DA-1 w.r.t auxiliary-input of non-restricted length.

(3) GMR-ZK for L under uniform DA-1 w.r.t auxiliary-input of restricted length.

(4) AIZK for L under uniform DA-1 w.r.t auxiliary-input of non-restricted length.

Proof: Combining Lemma 5.2, 5.8, 5.9, 5.11, and 5.12, Theorem 5.1 follows. ∎

Combining Theorem 3.1 and 5.1, it follows that there exists a 3-round ZK interactive argument for any NP language under non-uniform DLA, non-uniform DA-2 and the above variants of uniform and non-uniform DA-1.

5.1 3R-ZK is an Argument

Lemma 5.2 [3R-ZK is an argument] Assume 3R-HVZK satisfies the properties B1 and B2. Then 3R-ZK is an argument for L under non-uniform DA-2 and non-uniform DLA.

Proof: We need to show that 3R-ZK satisfies both the efficient completeness and the computational soundness. The efficient completeness is trivially satisfied because of the property B1 of $(\bar{P}, \bar{V})$. Therefore, we focus on the proof of the computational soundness.

Assume that 3R-ZK does not have negligible error probability in the computational soundness. Then there exist a cheating prover $\widehat{P}$, a polynomial $p_0(\cdot)$ and an infinite set $G = \{(x, w)\}$ of common inputs not in L and auxiliary-inputs such that for every $(x, w) \in G$, we have

$$\mathsf{Acc}\langle \widehat{P}(x, w), V(x, -)\rangle > 1/p_0(|x|).$$

Roughly speaking, G is a set of common inputs and auxiliary-inputs violating the computational soundness. In order to complete the proof, we will show that this contradicts non-uniform DLA under non-uniform DA-2.

Without loss of generality, we may assume that $\widehat{P}$ is deterministic since $\widehat{P}$ is allowed to take a non-uniform auxiliary-input w. Therefore, we may assume that the first message sent by $\widehat{P}$ is predetermined for each $(x, w) \in G$ (i.e., the auxiliary-input w may include it). From now on, we denote by $(M_1, p, q, g, A = g^a)$ the predetermined first message sent by $\widehat{P}$ on inputs (x, w). For simplicity, we also assume that (p, q, g) is in L_{PQG} since the probability that $\widehat{P}$ succeeds to cheat V is negligible for any $(p, q, g) \notin L_{PQG}$.

Let K be the set of all integers n for which G contains a common input x of length n. We construct an inverter $I = \{I_n\}_{n \in K}$ such that for all sufficiently large $n \in K$, there exists a triplet $(p, q, g) \in L_{PQG}$ such that p is of length n and

$$\Pr[a \leftarrow Z_q : I_n(p, q, g, g^a) = a]$$

is overwhelming. Before describing the inverter I, we claim that under non-uniform DA-2, for any (deterministic) cheating prover $\widehat{P}$, there exists a polynomial-size circuit family $\widehat{P}'$ who outputs not only (C, Y) but also c in Step P2. I uses $\widehat{P}'$ as a subroutine. Formally, the claim is as follows:

Claim 5.3 Assume non-uniform DA-2 holds. Then for any polynomial-time machine $\widehat{P}$ (the cheating prover), there exists a polynomial-size circuit family $\widehat{P}' = \{\widehat{P}'_n\}_{n \geq 1}$ which takes as input $(p, q, g, g^a, g^b, g^{ab})$ and outputs (M_2, C', Y', c) satisfying the following two conditions.

1. For every x and every w of polynomial length in $|x|$, it holds that

$$\Pr\left[\begin{array}{l} b \leftarrow Z_q; \\ (M1, p, q, g, g^a) = \widehat{P}(x, w, -); \\ (M_2, C, Y) = \widehat{P}(x, w, ((M1, p, q, g, g^a), (g^b, g^{ab}))); \\ (M_2, C', Y', c) = \widehat{P}'_n(p, q, g, g^a, g^b, g^{ab}) \end{array} : (M_2, C', Y') = (M_2, C, Y)\right] = 1.$$

2. For every polynomial $poly(\cdot)$, all sufficiently large n's, every x of length n and every w of polynomial length in n,

$$\Pr\left[\begin{array}{l} b \leftarrow Z_q; \\ (M1, p, q, g, g^a) = \widehat{P}(x, w, -); \\ (M_2, C', Y', c) = \widehat{P}'_n(p, q, g, g^a, g^b, g^{ab}) \end{array} : Y' = C'^b \ \wedge \ Y' \neq (g^b)^c \ \wedge \ Y' \neq (g^{ab})^c\right] < \frac{1}{poly(n)}.$$

Proof: We focus on the computation of the prover $\widehat{P}$ in P2, where the inputs are 9 strings $(x, w, M_1, p, q, g, A = g^a, B = g^b, X = g^{ab})$. On the other hand, in non-uniform DA-2, the inputs to M are only 6 strings $(p, q, g, A = g^a, B = g^b, X = g^{ab})$. However, we may consider that the 3 extra inputs (x, w, M_1) are incorporated into M_n. Therefore, we can apply non-uniform DA-2 to the computation of $\widehat{P}$ so that there exists the above circuit family $\widehat{P}'$. ▮

Now we describe the inverter $I = \{I_n\}_{n \in K}$ which uses $\widehat{P}'$ as a subroutine to solve the discrete logarithm problems. For simplicity, we describe the circuit I_n as a probabilistic polynomial-time machine with non-uniform advice strings of polynomial length in n. The random coins used can be eliminated later by the non-uniformity.

Circuit: $I_n (n \in K)$.

Input: (p, q, g) and $\beta (= g^\alpha)$, where (p, q, g) is a part of the first message sent by $\widehat{P}$ on inputs $(x, w) \in G$ and (x, p) are of length n.

Advice: $(M_1, A = g^a, a)$ and the description of the circuit $\widehat{P}'_n$.

Output: The discrete logarithm α such that $\beta = g^\alpha$.

Step 1: I_n generates a random number $b \in Z_q$ and computes $(B, X) = (\beta g^b, (\beta g^b)^a)$.

Step 2: I_n runs $\widehat{P}'_n$ on (p, q, g, A, B, X) to get its response (M_2, C, Y, c). I_n checks whether (M_1, Y, M_2) is accepting according to $(\bar{P}, \bar{V})$. If this is false, I_n goes back to Step 1, otherwise I_n goes to Step 3.

Step 3: I_n checks whether $Y = B^c$ or $Y = X^c$. If this is false, I_n goes back to Step 1, otherwise I_n goes to Step 4.

Step 4: I_n generates a random number $b' \in Z_q$ such that $b' \neq b$ and computes $(B', X') = (g^{b'}, (g^{b'})^a)$. If $B = B'$ then I_n stops and outputs $\alpha = b' - b \bmod q$ (Note that $\beta g^b = g^{b'}$). Otherwise, I_n goes to Step 5.

Step 5: As in Step 2, I_n runs $\widehat{P}'_n$ on (p, q, g, A, B', X') to get its response (M'_2, C', Y', c'). I_n checks whether (M_1, Y', M'_2) is accepting according to $(\bar{P}, \bar{V})$. If this is false, I_n goes back to Step 4, otherwise I_n goes to Step 6.

Step 6: I_n checks whether $Y' = B'^{c'}$ or $Y' = X'^{c'}$. If this is false, I_n goes back to Step 4, otherwise I_n goes to Step 7.

Step 7: I_n outputs α such that $\beta = g^\alpha$.

Intuitively, I_n tries to find two different accepting conversations in Step 1-6. Once they are found, I_n can easily compute α such that $\beta = g^\alpha$ in Step 7. Note that in Step 7, it holds that $Y = \beta^{ac} g^{abc}$ or $\beta^c g^{bc}$ and it also holds that $Y' = g^{ab'c'}$ or $g^{b'c'}$. Furthermore, Y must equal Y' by the strong soundness B2 of $(\bar{P}, \bar{V})$ since $x \notin L$. As a result, it is easy to computes α from the values (a, b, b', c, c').

Lastly, we need to analyze the expected running time of I_n. Recall that we assumed that

$$\mathsf{Acc}\langle \widehat{P}(x, w), V(x, -)\rangle > 1/p_0(|x|).$$

By the first condition of Claim 5.3, the probability that I_n passes the checks in Step 2 and 5 is at least $1/p_0(n)$. Furthermore, the second condition of Claim 5.3 guarantees that the probability that I_n passes the check in Step 3 (resp., Step 6) under the condition that I_n has already passed Step 2 (resp., Step 5) is negligible for all sufficiently large $n \in K$. Therefore, the expected running time of I_n is $O(p_0(n) \cdot \sum_{i=1}^{6} t_i(n) + t_7(n))$ where $t_i(n)$ is a polynomial bounding the running time of Step i.

As a result, we conclude that for all sufficiently large $n \in K$, there exists a triplet $(p, q, g) \in L_{PQG}$ such that p is of length n and

$$\Pr[\alpha \leftarrow Z_q : I_n(p, q, g, g^\alpha) = \alpha]$$

is overwhelming. This contradicts non-uniform DLA. ▮

5.2 3R-ZK is Zero-Knowledge

We first give a general lemma (Lemma 5.4) in the non-uniform model and then give two lemmas (Lemma 5.8 and 5.9) as special cases.

Lemma 5.4 Let $l_{ZK\_AI}(\cdot)$ be a polynomial. Let $l_{x,M_1}(\cdot)$ denote a polynomial bounding the length of (x, M_1). We denote by $l_{DA\_AI}(\cdot)$ the polynomial such that $l_{DA\_AI}(n) = l_{ZK\_AI}(n) + l_{x,M_1}(n)$. Assume that 3R-HVZK satisfies the properties B0 and B3. Then 3R-ZK is non-uniform zero-knowledge w.r.t. auxiliary-input of length l_{ZK_AI} for L under non-uniform DA-1 w.r.t. auxiliary-input of length l_{DA_AI}.

Proof: Firstly, we apply non-uniform DA-1 w.r.t. auxiliary-input of length l_{DA_AI} to the computation of the cheating verifier $\widehat{V}$ in V1.

Claim 5.5 Assume that non-uniform DA-1 w.r.t. auxiliary-input of length l_{DA_AI} holds. Then for every polynomial-size circuit family $\widehat{V} = \{\widehat{V}_n\}_{n\geq 1}$ (the cheating verifier), there exists another polynomial-size circuit family $\widehat{V}' = \{\widehat{V}'_n\}_{n\geq 1}$ which, on input $(x, y, M_1, p, q, g, g^a)$, outputs (B', X', b) satisfying the following two conditions.

1. For every $x \in L$, every $w \in W_L(x)$ and every $y \in \{0,1\}^{l_{ZK\_AI}(|x|)}$, it holds that

$$\Pr\left[\begin{array}{l}(M_1, p, q, g, g^a) \leftarrow P(x, w, -); \\ (B, X) = \widehat{V}_n(x, y, (M_1, p, q, g, g^a)); \qquad : (B', X') = (B, X) \\ (B', X', b) = \widehat{V}'_n(x, y, (M_1, p, q, g, g^a))\end{array}\right] = 1.$$

2. For every polynomial $poly(\cdot)$, all sufficiently large n's, every $x \in L$ of length n, $w \in W_L(x)$ and every $y \in \{0,1\}^{l_{ZK\_AI}(|x|)}$,

$$\Pr\left[\begin{array}{ll}(M_1,p,q,g,g^a) \leftarrow P(x,w,-); & \\ (B,X) = \widehat{V}_n(x,y,(M_1,p,q,g,g^a)); & : X' = B'^a \ \wedge \ B' \neq g^b \\ (B',X',b) = \widehat{V}'_n(x,y,(M_1,p,q,g,g^a)) & \end{array}\right] < \frac{1}{poly(n)}.$$

Proof: In the computation of $\widehat{V}$, the inputs are x, y, M_1, p, q, g and A. If we consider (x, y, M_1) as the auxiliary-input of length $l_{DA\_AI}(n)$ in non-uniform DA-1 w.r.t auxiliary-input of length l_{DA_AI} then the claim clearly holds. ▮

As described in the observation of the property R2 (in Section 3.3), we can construct the simulator $S_{\widehat{V}} = \{S_n\}_{n\geq 1}$ which uses $\widehat{V}' = \{\widehat{V}'_n\}_{n\geq 1}$ as a subroutine so that it is able to get the secret coins b used by $\widehat{V}$ and complete the simulation using b. $S_{\widehat{V}}$ also uses the honest-verifier simulator S_{HV} for $(\bar{P}, \bar{V})$ as a subroutine. For simplicity, we describe $S_{\widehat{V}}$ as a probabilistic polynomial-time machine with a non-uniform advice string of polynomial length in n.

Circuit: S_n.

Input: (x, y), where $x \in L$ is of length n and y is of length $l_{ZK\_AI}(n)$.

Advice: The description of the circuit $\widehat{V}'_n$.

Output: The view $[x, y, (M_1, p, q, g, A)(B, X)(M_2, C, Y); -]$ of $\widehat{V}_n$.

Step 1: S_n runs S_{HV} on input x to get the view of $\bar{V}$

$$[x, -, ((p, q, g, M_1), Y, M_2); R_Y],$$

where Y is an element of order q in Z_p^*.

Step 2: S_n generates a uniformly at random in Z_q and computes $A = g^a$.

Step 3: S_n runs $\widehat{V}'(x, y, (M_1, p, q, g, A))$ to get (B, X, b).

Step 4: S_n checks whether $X = B^a$. If this is true, S_n goes to Step 5, otherwise, S_n stops and outputs $[x, y, (M_1, p, q, g, A)(B, X); -]$.

Step 5: S_n checks whether $B = g^b$. If this is true, S_n goes to Step 6, otherwise, S_n aborts.

Step 6: S_n computes C such that $Y = C^b$.

Step 7: S_n outputs $[x, y, (M_1, p, q, g, A)(B, X)(M_2, C, Y); -]$.

Clearly, S_n runs in probabilistic polynomial-time. Moreover, the second condition of Claim 5.5 guarantees that the probability that $S_{\widehat{V}}$ aborts in Step 5 is negligible.

In order to complete the proof, we show that the simulator $S_{\widehat{V}}$ outputs a view computationally indistinguishable from the real view between P and $\widehat{V}$.

Claim 5.6 The distribution ensemble $\{S_{\widehat{V}}(x)\}_{x\in L, y\in\{0,1\}^{l_{ZK\_AI}(|x|)}}$ is computationally indistinguishable from $\{\langle P(x,w), \widehat{V}(x,-)\rangle\}_{x\in L, w\in W_L(x), y\in\{0,1\}^{l_{ZK\_AI}(|x|)}}$.

Proof: The proof is by contradiction. Assume there exists a distinguisher D that can distinguish

$$\{S_{\widehat{V}}(x,y)\}_{x\in L, y\in\{0,1\}^{l_{ZK\_AI}(|x|)}} \text{ from } \{\langle P(x,w), \widehat{V}(x,-)\rangle\}_{x\in L, w\in L_W(x), y\in\{0,1\}^{l_{ZK\_AI}(|x|)}}.$$

Then we can construct a distinguisher D' that distinguishes the distribution ensemble

$$\{S_{HV}(x)\}_{x\in L} \text{ from } \{\langle \bar{P}(x,w), \bar{V}(x,-)\rangle\}_{x\in L, w\in W_L(x)}$$

in contradiction to the property B3 of $(\bar{P}, \bar{V})$. Given $\bar{V}$'s view $[x, -, ((p,q,g,M_1), Y, M_2); R_Y]$ (from either distribution ensemble), D' extends it to a view of $\widehat{V}$ and invokes D on the extended view as follows:

Distinguisher: $D' = \{D'_{x,y}\}_{x\in L, y\in\{0,1\}^{l_{ZK\_AI}(|x|)}}$.
Input: $[x, -, ((p,q,g,M_1), Y, M_2); R_Y]$.
Output: 0 or 1.
Step 1: D'_n runs Step 2-7 of S_n to extend the input $[x, -, ((p,q,g,M_1), Y, M_2); R_Y]$ to a view of $\widehat{V}$

$$[x, y, ((M_1, p, q, g, A), (B, X), (M_2, C, Y)); -],$$

where the input $((p,q,g,M_1), Y, M_2)$ is used as a result of Step 1 in S_n.
Step 2: D'_n invokes D_n on this extended view.

If the input is from $\{\langle \bar{P}(x,w), \bar{V}(x,-)\rangle\}_{x\in L, w\in W_L(x)}$, the extended view is distributed statistically close to $\{\langle P(x,w), \widehat{V}(x,-)\rangle\}_{x\in L, y\in\{0,1\}^{l_{ZK\_AI}(|x|)}}$ because of the following two facts:

1. The probability that $S_{\widehat{V}}$ aborts in Step 5 is negligible.
2. The output distributions of $\widehat{V}$ and $\widehat{V}'$ are identical on (B, X) by the first condition of Claim 5.5.

On the other hand, if the input is from $\{S_{HV}(x)\}_{x\in L}$, the extended view is clearly identical to the distribution ensemble $\{S_{\widehat{V}}(x,y)\}_{x\in L, y\in\{0,1\}^{l_{ZK\_AI}(|x|)}}$.

Therefore, D' distinguishes the two distribution ensembles

$$\{S_{HV}(x)\}_{x\in L} \text{ and } \{\langle \bar{P}(x,w), \bar{V}(x,-)\rangle\}_{x\in L, w\in W_L(x)}$$

in contradiction to the property B3 of $(\bar{P}, \bar{V})$. ▮

Lemma 5.4 follows. ▮

Remark 5.7 Claim 5.5 does not say that $\widehat{V}'$ can be effectively constructed given $\widehat{V}$, but rather that it exists. However, it is sufficient since the definition of NUZK w.r.t. auxiliary input of length l_{ZK_AI} also does not require that $S_{\widehat{V}}$ can be effectively constructed given $\widehat{V}$.

The following lemmas are easily obtained as special cases of Lemma 5.4.

Lemma 5.8 [3R-ZK is NUZK] Assume that 3R-HVZK satisfies the properties B0 and B3. Then 3R-ZK is non-uniform zero-knowledge under non-uniform DA-1 w.r.t. auxiliary-input of restricted length.

Proof: Consider the special case of $l_{ZK\_AI} = 0$ in Lemma 5.4. For 3R-ZK to achieve NUZK, it is sufficient to assume non-uniform DA-1 w.r.t. auxiliary-input of length l_{x,M_1}, which is a pre-specified polynomial. ▮

Lemma 5.9 [3R-ZK is AINUZK] Assume that 3R-HVZK satisfies the properties B0 and B3. Then 3R-ZK is auxiliary-input non-uniform zero-knowledge under non-uniform DA-1 w.r.t. auxiliary-input of non-restricted length.

Proof: Consider the special case that l_{ZK_AI} is an arbitrary polynomial in Lemma 5.4. In this case, l_{DA_AI} ranges over the set of all polynomials. Therefore, for 3R-ZK to achieve AINUZK, we require non-uniform DA-1 w.r.t. auxiliary-input of non-restricted length. ▮

The above results easily extend to the uniform model. We give the results in the uniform model without the proofs.

Lemma 5.10 Let $l_{ZK\_AI}(\cdot)$ be a polynomial. Let $l_{x,M_1}(\cdot)$ denote a polynomial bounding the length of (x, M_1). We denote by $l_{DA\_AI}(\cdot)$ the polynomial such that $l_{DA\_AI}(n) = l_{ZK\_AI}(n) + l_{x,M_1}(n)$. Assume that 3R-HVZK satisfies the properties B0 and B3. Then 3R-ZK is zero-knowledge w.r.t. auxiliary-input of length l_{ZK_AI} for L under uniform DA-1 w.r.t. auxiliary-input of length l_{DA_AI}.s

Lemma 5.11 [3R-ZK is GMR-ZK] Assume that 3R-HVZK satisfies the properties B0 and B3. Then 3R-ZK is GMR zero-knowledge under uniform DA-1 w.r.t. auxiliary-input of restricted length.

Lemma 5.12 [3R-ZK is AIZK] Assume that 3R-HVZK satisfies the properties B0 and B3. Then 3R-ZK is auxiliary-input zero-knowledge under uniform DA-1 w.r.t. auxiliary-input of non-restricted length.

6 Discussion

In this section, we discuss the properties and the implications of our results. First, we comment that the augmented versions of DA-1 have some kind of reverse-engineering property. Next, we show how hard it is to prove that $Cl(AIZK)$ equals $Cl(BSZK)$. Lastly, we argue that our transformation preserve the statistical zero-knowledgeness.

6.1 Reverse-Engineering Property

In order to demonstrate the ZK property of 3R-ZK, we required several augmentations of DA-1. Unfortunately, all of them have some kind of reverse-engineering property which seems to be unreasonable [Go98b].

Take for example non-uniform DA-1 w.r.t. auxiliary-input of length l_{DA_AI}. Consider $M = \{M_n\}_{1 \geq n}$ as a *universal* polynomial-size circuit family which takes as its auxiliary-input of length $l_{DA\_AI}(n)$ the description of a program Π of size $l_{DA\_AI}(n)$ which, given (p, q, g, g^a), outputs (B, X) such that $X = B^a$. Then, non-uniform DA-1 w.r.t. auxiliary-input of length l_{DA_AI} says that there exists another circuit family M' which can *reverse-engineer* any program Π of size $l_{DA\_AI}(n)$ and output b used by Π. In particular, non-uniform DA-1 w.r.t. auxiliary-input of non-restricted length, which is required for AINUZK, says that a polynomial-size circuit family M' can reverse-engineer any program Π of any polynomial size. Also, non-uniform DA-1 w.r.t. auxiliary-input of restricted length, which is required for NUZK, says a polynomial-size circuit family M' can reverse-engineer any program Π whose size is bounded by a pre-specified polynomial size l_{x,M_1}. The former property seems to be stronger than the latter.

In the case of uniform DA-1 w.r.t. auxiliary-input of length l_{DA_AI}, the reverse-engineering property is stronger than in the non-uniform case. In particular, uniform DA-1 w.r.t. auxiliary-input of non-restricted length, which is required for AIZK, says that a probabilistic polynomial-time machine M' can reverse-engineer any program Π of any polynomial size. Therefore, we may say that this assumption seems to have the strongest reverse-engineering property and that our result regarding AIZK (Lemma 5.12) seems to be most problematic.

6.2 $Cl(AIZK)$ v.s. $Cl(BSZK)$

As described in [GoOr94], it is an open problem whether or not it holds $Cl(AIZK) \subset Cl(BSZK)$. Our results imply an answer to this problem.

Corollary 6.1 *Under uniform DA-1 w.r.t. auxiliary-input of non-restricted length, it holds that* $Cl(AIZK) \subset Cl(BSZK)$.

Proof: Combining our results with the results of [GoKr96], it follows. ▮

Although the required assumption is unreasonable, it has an interesting implication. That is, proving the equality $Cl(AIZK) = Cl(BSZK)$ is as hard as disproving uniform DA-1 w.r.t. auxiliary-input of non-restricted length. We believe that it is hard to disprove it by the standard technique. Therefore, Corollary 6.1 can be viewed as an evidence that it is hard to prove that $Cl(AIZK) = Cl(BSZK)$.

6.3 Statistical Zero-Knowledgeness

Our transformation preserves the statistical zero-knowledgeness. That is, if the starting protocol 3R-HVZK satisfies the statistical zero-knowledgeness with respect to honest verifier in the property B3, then 3R-ZK satisfies statistical-zero-knowledgeness as well. The class of languages satisfying the above hypothesis is indeed very large although we don't know whether it is as large as NP languages. It includes the following languages.

1. Random-self-reducible languages [TW87].

2. All monotone formulae over random-self-reducible languages [DDPY94].

3. All languages having a non-interactive statistical zero-knowledge interactive proof [DDP94].

We note that the perfect zero-knowledgeness can not be preserved since we can not get rid of the possibility that the simulator aborts in Step 5 although it happens with negligible probability.

7 Concluding Remarks

We provided a non-black-box simulation technique based on a non-standard computational assumption so that we could construct a 3-round ZK protocol for any NP language. However, the required computational assumptions have the reverse-engineering property which seems to be unreasonable.

Therefore, it is still open whether one can construct a 3-round zero-knowledge protocol for a non-trivial language based on reasonable computational assumptions. We believe that it is interesting to resolve the situation either way: Either provide a plausible construction based on reasonable computational assumptions, or prove that some kind of reverse-engineering property is essential for 3-round ZK protocols for non-trivial languages.

Acknowledgments

This paper owes much to the valuable comments and suggestions of Oded Goldreich. He kindly helped us to revise an earlier version of this paper. We would like to thank Giovanni Di Crescenzo for allowing us to use his observations about statistical zero-knowledgeness in Section 6.3. We would like to thank the anonymous referees of CRYPTO'98 for valuable comments. We also thank Masahiro Wada, Koji Nakao, and Kenji Suzuki for their encouragement.

References

[BeGo92] M. Bellare and O. Goldreich , "On Defining Proofs of Knowledge, " Proceedings of CRYPTO'92, 1992.

[BeRo93] M. Bellare and P. Rogaway, "Random Oracles are Practical: a paradigm for designing efficient protocols, " Proceedings of the 1st ACM Conference on Computer and Communications Security, pp. 62-73, 1993.

[BJY97] M. Bellare, M. Jakobsson and M. Yung, "Round-Optimal Zero-Knowledge Arguments Based on any One-Way Function, " Proceedings of Eurocrypt' 97, 1997.

[BMO90] M. Bellare, S. Micali and R. Ostrovsky, " Perfect Zero-Knowledge in Constant Rounds," Proceedings of 22nd STOC, 1990.

[Bl86] M. Blum, "How to Prove a Theorem So No One Else Can Claim It," Proceedings of the International Congress of Mathematicians, pp.1444-1451, 1986.

[BCC88] G. Brassard, D. Chaum and C. Crépeau, "Minimum Disclosure Proofs of Knowledge, " Journal of Computer and System Sciences, Vol. 37, No. 2, pp. 156-189, 1988.

[BCY89] G. Brassard, C. Crépeau and M. Yung, " Everything in NP Can Be Argued in Perfect Zero-Knowledge in a Bounded Number of Rounds, " Proceedings of 16th ICALP, pp.123-136, 1989.

[BrCr86] G. Brassard and C. Crépeau, "Non-Transitive Transfer of Confidence : A Perfect Zero-Knowledge Interactive Protocol for SAT and Beyond, " Proceedings of 27th FOCS, 1986.

[Da91] I. Damgård, "Towards Practical Public Key Systems Secure Against Chosen Ciphertext Attacks," Proceedings of CRYPTO'91, pp.445-456, 1991.

[DDP94] A. De Santis, G. Di Crescenzo, and G. Persiano, "The Knowledge Complexity of Quadratic Residuosity Languages," Theoretical Computer Science, vol. 132, pp. 291–317, 1994.

[DDPY94] A. De Santis, G. Di Crescenzo, G. Persiano and M. Yung, "On monotone formula closure of Statistical Zero-Knowledge," Proceedings of 35th FOCS, pp. 454–465, 1994.

[DH76] W. Diffie and M. Hellman, "New Directions in Cryptography," IEEE Trans. Inform. Theory, Vol.22, No.6, pp.644-654, 1976.

[FeSh89] U. Feige and A. Shamir, "Zero Knowledge Proofs of Knowledge in Two Rounds," Proceedings of CRYPTO'89, pp.526-544, 1989.

[FeSh90] U. Feige and A. Shamir, "Witness Indistinguishable and Witness Hiding Protocols," Proceedings of 22nd STOC, pp.416-426, 1990.

[Go93] O. Goldreich, "A Uniform-Complexity Treatment of Encryption and Zero-Knowledge," Journal of Cryptology, Vol.6, No. 1, pp.21-53, 1993.

[Go98a] O. Goldreich, "Foundations of Cryptography (Fragments of a Book - Version 2.03)," February 27, 1998.

[Go98b] O. Goldreich, private communication, May 1998.

[GoKa96] O. Goldreich and A. Kahan, "How to Construct Constant-Round Zero-Knowledge Proof Systems for NP," Journal of Cryptology, Vol.9, No. 3, pp.167-190, 1996.

[GoKr96] O. Goldreich and H. Krawczyk, "On the Composition of Zero-Knowledge Proof Systems," SIAM Journal on Computing, Vol.25, No.1, pp.169-192, 1996.

[GMW91] O. Goldreich, S. Micali, and A. Wigderson, "Proofs that Yield Nothing But Their Validity or All Languages in NP Have Zero-Knowledge Proof Systems," Journal of the ACM, Vol.38, No.1, pp.691-729, 1991.

[GoOr94] O. Goldreich and Y. Oren, "Definitions and Properties of Zero-Knowledge Proof Systems," Journal of Cryptology, Vol.7, No. 1, pp.1-32, 1994.

[GMR85] S. Goldwasser, S. Micali, and C. Rackoff, "The Knowledge Complexity of Interactive Proofs," Proceedings of 17th STOC, pp.291-304, 1985.

[GMR89] S. Goldwasser, S. Micali, and C. Rackoff, "The Knowledge Complexity of Interactive Proof Systems," SIAM Journal on Computing, Vol.18, No.1, pp.186-208, 1989.

[HT98] S. Hada and T. Tanaka, "On the Existence of 3-Round Zero-Knowledge Protocols, " , Proceedings of CRYPTO'98, pp. 408-423, 1998.

[Ra80] M. O. Rabin, "Probabilistic Algorithm for Testing Primality, " Journal of Number Theory, Vol. 12, pp.128-138, 1980.

[SS77] R. Solovay and V. Strassen, "A Fast Monte-Carlo Test for Primality," SIAM Journal on Computing, Vol.6, No.1, pp.84-86, 1977.

[TW87] M. Tompa and H. Woll, "Random Self-Reducibility and Zero Knowledge Interactive Proofs of Possession of Information," Proceedings of 28th FOCS, pp.472-482, 1987.

A Blum's Zero-Knowledge Protocol for the Hamiltonian Circuit Problem

The following protocol is an atomic ZK interactive proof for the Hamiltonian circuit problem [Bl86][Go98a].

Protocol: 3R-HVZK for the Hamiltonian Circuit Problem.

Common Input: a graph $G = (V, E)$ of size n, $V = \{1, 2, \cdots, |V|\}$.

Prover's Aux-Input: a permutation ϕ over V representing the order of vertices along a Hamiltonian circuit, i.e., $\phi(1)$,$\phi(2)$,$\cdots$,$\phi(|V|)$,$\phi(1)$ represent a Hamiltonian circuit.

P1: The prover chooses a random permutation π over V and generates an isomorphic copy $G' = \pi(G) = (V, E')$. For each pair $(j, k) \in V^2$, p_i sets $e_{j,k} = 1$ if $(j, k) \in E'$ and $e_{j,k} = 0$ otherwise. The prover computes a commitment $x_{j,k} = E(e_{j,k}, r_{j,k})$ for each $e_{j,k}$ and sends all the $x_{j,k}$'s to the verifier. $E(\cdot, \cdot)$ is a non-uniformly secure encryption function and used for a bit-commitment scheme. Let m_1 denote this first message.

V1: The verifier selects a random bit y and sends it to the prover.

P2: If the prover receives $y = 0$, the prover reveals all the commitments $x_{j,k}$'s, i.e., sends $e_{j,k}, r_{j,k}$ for each j, k and π to the verifier. If the prover receives $y = 1$, the prover reveals only $|V|$ commitments, specifically those corresponding to the Hamiltonian circuit in G', i.e., sends $\phi'(\cdot) = \pi(\phi(\cdot))$ and $r_{\phi'(1),\phi'(2)}$, $r_{\phi'(2),\phi'(3)}$,$\cdots$, $r_{\phi'(|V|),\phi'(1)}$ to the verifier. Let m_2^0 and m_2^1 denote the messages in the case of $y = 0$ and $y = 1$, respectively.

V2: In case of $y = 0$, the verifier accepts if all the commitments are revealed correctly and indeed correspond to the graph $\pi(G)$. In case of $y = 1$, the verifier accepts if it holds that $x_{\phi'(1),\phi'(2)} = E(1, r_{\phi'(1),\phi'(2)})$, $x_{\phi'(2),\phi'(3)} = E(1, r_{\phi'(2),\phi'(3)})$, $\cdots$, $x_{\phi'(|V|),\phi'(1)} = E(1, r_{\phi'(|V|),\phi'(1)})$.

The above atomic protocol has constant error probability $\frac{1}{2}$. The following parallel composition is used to decrease the error probability while maintaining the number of rounds.

P1: The prover sends $M_1 = m_{1,1}, m_{1,2}, \cdots, m_{1,n}$.

V1: The verifier sends $Y = y_1, y_2, \cdots, y_n$.

P2: The prover sends $M_2 = m_{2,1}^{y_1}, m_{2,2}^{y_2}, \cdots, m_{2,n}^{y_n}$

V2: The verifier accepts if and only if all the responses in M_2 are valid.

In the above, the message Y is chosen uniformly at random in $\{0, 1\}^n$, but it may be chosen from any polynomially sampleable subdomain of $\{0, 1\}^n$.

B A Flaw in CRYPTO'98 version [HT98]

We wrongly identified the following assumptions:

- non-uniform DA-1 (called SDHA-1 in [HT98]).
- non-uniform DA-1 w.r.t. auxiliary-input of restricted length.
- non-uniform DA-1 w.r.t. auxiliary-input of non-restricted length.

As a result, we wrongly claimed that 3R-ZK achieves the notion of AINUZK under non-uniform DA-1 (Lemma 12 in [HT98]).