J. Cryptology (1997) 10: 37–49

Journal of
CRYPTOLOGY

A Language-Dependent Cryptographic Primitive

Toshiya Itoh and Yuji Ohta
Department of Information Processing,
Interdisciplinary Graduate School of Science and Engineering,
Tokyo Institute of Technology, 4259 Nagatsuta, Midori-ku,
Yokohama 226, Japan
titoh@ip.titech.ac.jp ohta@ip.titech.ac.jp

Hiroki Shizuya
Education Center for Information Processing, Tohoku University,
Kawauchi, Aoba-ku, Sendai 980-77, Japan
shizuya@ecip.tohoku.ac.jp

Communicated by Joan Feigenbaum

Received 19 July 1994 and revised 25 August 1995

Abstract. In this paper we provide a new cryptographic primitive that generalizes several existing zero-knowledge proofs and show that if a language L induces the primitive, then there exists a perfect zero-knowledge proof for L. In addition, we present several kinds of languages inducing the primitive, some of which are not known to have a perfect zero-knowledge proof.

Key words. Bit commitments, Zero-knowledge proofs, Language membership, Proofs of knowledge.

1. Introduction

1.1. *Background and Motivation*

A bit commitment is a two-party (interactive) protocol between a sender S and a receiver R in which after the sender S commits to a bit $b \in \{0, 1\}$ at hand, (1) the sender S cannot change his mind; and (2) the receiver R learns nothing about the value of the bit b. Bit commitments have diverse applications to cryptographic protocols, especially to zero-knowledge proofs (see, e.g., [10], [8], [19], [13], and [3]). According to the computational power of senders and receivers, bit commitments can be classified into the four possible types shown in Table 1.

Feige and Shamir [10] used a bit commitment of Type A to show that any language $L \in \mathcal{NP}$ has a two-round perfect zero-knowledge argument (or computationally sound proof) whose protocol is a proof of knowledge. Brassard *et al.* [8] and Naor *et al.* [19] showed that any language $L \in \mathcal{NP}$ has a perfect zero-knowledge argument assuming

Table 1. Classification of bit commitments.

	Computational power of sender S	Computational power of receiver R
Type A	Polynomial-time bounded	Polynomial-time bounded
Type B	Polynomial-time bounded	Computationally unbounded
Type C	Computationally unbounded	Polynomial-time bounded
Type D	Computationally unbounded	Computationally unbounded

the existence of a bit commitment of Type B and Bellare *et al.* [3] showed that any honest verifier statistical zero-knowledge proof for a language L can be transformed to a statistical zero-knowledge proof for the language L assuming the existence of a bit commitment of Type B. Indeed, Naor *et al.* [19] showed that a bit commitment of Type B with *simulatable* property can be constructed from any oneway permutation and Bellare *et al.* [3] showed that a bit commitment of Type B with *chameleon* property can be constructed from the certified discrete logarithm. In addition, Goldreich *et al.* [13] used a bit commitment of Type C to show that any language $L \in \mathcal{NP}$ has a computational zero-knowledge proof.

For technical reasons, we assume that a bit commitment f is noninteractive, i.e., (1) to commit to a bit $b \in \{0, 1\}$, the sender S randomly chooses $r \in \{0, 1\}^k$ and sends $C = f(b, r)$ to the receiver R; and (2) to decommit to the bit b, S reveals $b \in \{0, 1\}$ and $r \in \{0, 1\}^k$ such that $C = f(b, r)$ and R checks that $C = f(b, r)$. We use $f(b)$ to denote the distribution over r for each b. Now we look at the properties required to noninteractive bit commitments.

Assume that the sender S is computationally unbounded. If there exist $r, s \in \{0, 1\}^k$ such that $f(0, r) = f(1, s)$, then a cheating sender S^* chooses r to compute $C = f(0, r)$ and reveals 1 and s to change his mind. Thus any r, s must satisfy that $f(0, r) \neq f(1, s)$. We refer to such a bit commitment f as *transparent*. Assume that the receiver R is computationally unbounded. If the distribution $f(0)$ is not (almost) identical to the distribution $f(1)$, i.e., $\sum_{\alpha \in \{0,1\}^*} |\Pr\{f(0, r) = \alpha\} - \Pr\{f(1, s) = \alpha\}|$ is not small, then a cheating receiver R^* might learn something about the value of the bit b only looking at $C = f(b, r)$. Thus the distributions $f(0)$ and $f(1)$ must be (almost) identical. Here we refer to such a bit commitment f as *opaque*. If both the sender S and the receiver R are computationally unbounded, then any bit commitment f must be transparent and opaque, however, it is impossible to implement such a bit commitment algorithmically [20]. This implies that there exists inherently no way of designing bit commitments of Type D. Thus the only possible way of doing this is to implement such a (noninteractive) bit commitment physically. This is referred to as an *envelope* [13]. Assuming the existence of the envelope, Goldreich *et al.* [13] showed that any language $L \in \mathcal{NP}$ has a perfect zero-knowledge proof and then Ben-Or *et al.* [4] showed that any language $L \in \mathcal{IP}$ has a perfect zero-knowledge proof.

There have been attempts to provide general frameworks to capture known zero-knowledge proofs of various kinds. The notion of random self-reducible [21] has been one of the most successful primitives. The goal of this paper is to construct algorithmically a bit commitment of Type D in a somewhat different setting and to provide an alternative framework that generalizes several existing zero-knowledge proofs under a common abstraction.

1.2. *Results*

In this paper we consider the following framework: Let $L \subseteq \{0, 1\}^*$ be a language. The function f_L is allowed to have an additional input $x \in \{0, 1\}^*$, and we let $f_L(x, b)$ be the distribution over $r \in \{0, 1\}^{k(|x|)}$ for each $b \in \{0, 1\}$. Informally, the function f_L is positively (resp. negatively) **opaque** if, for every $x \in L$ (resp. $x \notin L$), the distribution $f_L(x, 0)$ is *identical* to the distribution $f_L(x, 1)$ and the function f_L is positively (resp. negatively) **transparent** if, for every $x \in L$ (resp. $x \notin L$), the distribution $f_L(x, 0)$ is *disjoint* from the distribution $f_L(x, 1)$.

We first present several examples of languages that induce positively opaque and negatively transparent functions. It should be noted that every known random self-reducible language, e.g., graph isomorphism, quadratic residuosity, multiplicative subgroup $\langle g \rangle_p$ of Z_p^*, etc., induces positively opaque and negatively transparent functions, but some examples of languages given in this paper might not be random self-reducible.

We then show that languages inducing positively opaque and negatively transparent functions have zero-knowledge proofs, i.e.,

Theorem 4.3. *If a language L induces a positively opaque and negatively transparent funtion, then there exists a prover-practical unbounded round perfect zero-knowledge proof for L.*

The prover-practical proof [7] is an interactive proof for a language $L \in \mathcal{NP}$ in which the honest prover P runs in probabilistic polynomial time provided some trapdoor information on input $x \in L$ is initially written on the private auxiliary tape of P. It is known that any random self-reducible language has a prover-practical bounded round perfect zero-knowledge proof [21], [2]. The notion of prover-practical is useful for applications. In particular, prover-practical zero-knowledge proofs for $\mathcal{NP}$-complete languages are desirable for practical purposes, however, some unproven assumptions are required to construct such proofs (computational zero-knowledge proofs) for $\mathcal{NP}$-complete languages (see, e.g., [5] and [13]). Thus Theorem 4.3 provides an alternative framework (to random self-reducible languages) to construct prover-practical perfect zero-knowledge proofs without any unproven assumption.

We finally show that languages inducing positively transparent and negatively opaque functions have zero-knowledge proofs, i.e.,

Theorem 4.5. *If a language L induces a positively transparent and negatively opaque function, then there exists a bounded round perfect zero-knowledge proof for L.*

Every language whose complement is known to be random self-reducible induces a positively transparent and negatively opaque function but the exmples of languages inducing positively transparent and negatively opaque functions include ones that do not seem to be random self-reducible. Thus Theorem 4.5 can be regarded as the generalization of the zero-knowledge proof for quadratic nonresiduosity [16] or graph nonisomorphism [13].

2. Preliminaries

Let $L \subseteq \{0, 1\}^*$ be a language and let k be a polynomial. Assume that $f_L(x, b, r)$ is a polynomial (in $|x|$) time computable function for any $b \in \{0, 1\}$ and any $r \in \{0, 1\}^{k(|x|)}$. We use $f_L(x, b)$ to denote the distribution over r for each b.

Definition 2.1. Let L be a language. A function f_L is positively (resp. negatively) *opaque* if, for each $x \in L$ (resp. $x \notin L$), $f_L(x, 0)$ is identical to $f_L(x, 1)$.

Definition 2.2. Let L be a language. A function f_L is positively (resp. negatively) *transparent* if, for each $x \in L$ (resp. $x \notin L$), there do not exist r, s such that $f_L(x, 0, r) = f_L(x, 1, s)$.

Definition 2.3. A language L induces a positively opaque and negatively transparent (resp. positively transparent and negatively opaque) function if there exists f_L that is positively opaque and negatively transparent (resp. positively transparent and negatively opaque).

The positively opaque and negatively transparent property guarantees that, for every $x \in L$, any all powerful cheating receiver R^* cannot guess better than at random the value of the bit $b \in \{0, 1\}$ after receiving a random point from the distribution $f_L(x, b)$ and, for every $x \notin L$, any all powerful cheating sender S^* cannot change his mind after sending any point from the distribution $f_L(x, b)$. From Definitions 2.1 and 2.2, it follows that, for any language L inducing a positively opaque and negatively transparent function, $x \in L$ iff there exist r, s such that $f_L(x, 0, r) = f_L(x, 1, s)$. Thus any language L inducing a positively opaque and negatively transparent function is in $\mathcal{NP}$.

Contrary to the positively opaque and negatively transparent property, the positively transparent and negatively opaque property guarantees that, for every $x \in L$, any all powerful cheating sender S^* cannot change his mind after sending any point from the distribution $f_L(x, b)$ and, for every $x \notin L$, any all powerful cheating receiver R^* cannot guess better than at random the value of the bit $b \in \{0, 1\}$ after receiving a random point from the distribution $f_L(x, b)$. From Definition 2.3, it is obvious that a language L induces a positively transparent and negatively opaque function iff $\bar{L}$ (the complement of L) induces a positively opaque and negatively transparent function. This implies that L is in co-$\mathcal{NP}$.

Definition 2.4 [16]. An interactive protocol $\langle P, V\rangle$ is an interactive proof for a language L if there exists a verifier V (called the honest verifier) that satisfies the following:

- *Completeness*: there exists a prover P (called the honest prover) such that, for every $k > 0$ and all but finitely many $x \in L$, $\langle P, V\rangle$ halts and accepts x with probability at least $1 - |x|^{-k}$, where the probabilities are taken over the coin tosses of P and V.
- *Soundness*: for every $k > 0$, all but finitely many $x \notin L$, and any prover P^*, $\langle P^*, V\rangle$ halts and accepts x with probability at most $|x|^{-k}$, where the probabilities are taken over the coin tosses of P^* and V (the prover when $x \notin L$ is usually called a cheating prover).

Note that P is computationally unbounded while V is probabilistic polynomial (in $|x|$) time.

For an interactive proof $\langle P, V\rangle$ on common input x, we use $\langle P, V\rangle(x)$ to denote the distribution over the coin tosses of P and V. For a probabilistic Turing machine M on input x, we use $M(x)$ to denote the distribution over the coin tosses of M. Now we present a formal definition of blackbox simulation zero-knowledge. In the rest of this paper we assume that a term "zero-knowledge" implies "blackbox simulation" zero-knowledge.

Definition 2.5 [14]. An interactive proof $\langle P, V\rangle$ for a language L is (blackbox simulation) *perfect* zero-knowledge if there exists a probabilistic polynomial-time Turing machine M such that, for any (cheating) verifier V^* and all but finitely many $x \in L$, the distribution $M(x; V^*)$ is *identical* to the distribution $\langle P, V^*\rangle(x)$, where $M(\cdot; A)$ denotes a Turing machine with blackbox access to a Turing machine A.

For practical purposes, Boyar *et al.* [7] defined a notion of *prover-practical* (zero-knowledge) interactive proof.

Definition 2.6 [7]. An interactive proof $\langle P, V\rangle$ for a language $L \in \mathcal{NP}$ is *prover-practical* if the honest prover P runs in probabilistic polynomial time provided some trapdoor information on input $x \in L$ is initially written on the private auxiliary tape of P.

For each language $L \in \mathcal{NP}$, we use ρ_L to denote a polynomial-time computable predicate that witnesses $L \in \mathcal{NP}$, i.e., $x \in L$ iff there exists w such that $\rho_L(x, w) = 1$. Let $A, B \in \mathcal{NP}$ and let g be a reduction from A to B, i.e., g is a polynomial-time computable function such that $x \in A$ iff $g(x) \in B$. Then the following is essential to show Theorems 4.3 and 4.5.

Definition 2.7. Let $A, B \in \mathcal{NP}$ and let ρ_A, ρ_B be the defining predicates of A, B, respectively. A reduction g from A to B is *witness-preserving* (with respect to ρ_A, ρ_B) if there exists a polynomial-time computable function h that given w such that $\rho_A(x, w) = 1$ for each $x \in A$, $h(x, w)$ satisfies that $\rho_B(g(x), h(x, w)) = 1$.

Definition 2.8. Let $A, B \in \mathcal{NP}$ and let ρ_A, ρ_B be the defining predicates of A, B, respectively. A reduction g from A to B is *polynomial-time invertible* (with respect to ρ_A, ρ_B) if there exists a polynomial-time computable function γ that given w' such that $\rho_B(g(x), w') = 1$ for each $x \in A$, $\gamma(g(x), w')$ satisfies that $\rho_A(x, \gamma(g(x), w')) = 1$.

3. Examples

It is obvious from Definition 2.3 that L induces a positively transparent and negatively opaque function iff $\bar{L}$ (the complement of L) induces a positively opaque and negatively transparent function. Thus we only exemplify several languages that induce positively opaque and negatively transparent functions.

Let $G = (V, E_G)$ and $H = (V, E_H)$ be graphs. We use $G \simeq H$ to imply that G is isomorphic to H, i.e., there exists a permutation π on V such that $(u, v) \in E_G$ iff $(\pi(u), \pi(v)) \in E_H$.

Definition 3.1. *Universal Graph Isomorphism Tuple* (UGIT) is the language of graph tuples.

$$\mathrm{UGIT} = \left\{ \langle h, \langle G_1^0, G_1^1\rangle, \langle G_2^0, G_2^1\rangle, \ldots, \langle G_h^0, G_h^1\rangle\rangle \,\middle|\, \bigwedge_{i=1}^{h} [G_i^0 \simeq G_i^1] \right\},$$

where h is a positive integer.

Definition 3.2. *Existential Graph Isomorphism Tuple* (EGIT) is the language of graph tuples.

$$\mathrm{EGIT} = \left\{ \langle h, \langle G_1^0, G_1^1\rangle, \langle G_2^0, G_2^1\rangle, \ldots, \langle G_h^0, G_h^1\rangle\rangle \,\middle|\, \bigvee_{i=1}^{h} [G_i^0 \simeq G_i^1] \right\},$$

where h is a positive integer.

It is obvious that UGIT and EGIT are graph isomorphism when $h = 1$.

Definition 3.3. cMODd is the language of integers N having the following property. If $N = p_1^{e_1} p_2^{e_2} \cdots p_h^{e_h}$ is the factorization of N, then $p_i \equiv c (\mathrm{mod}\, d)$ for each i $(1 \le i \le h)$.

In the following we show that the languages UGIT, EGIT, and 1MOD4 induce positively opaque and negatively transparent functions f_{UGIT}, f_{EGIT}, and f_{1MOD4}, respectively.

Proposition 3.4. UGIT *induces a positively opaque and negatively transparent function.*

Proof. For $x = \langle h, \langle G_1^0, G_1^1\rangle, \langle G_2^0, G_2^1\rangle, \ldots, \langle G_h^0, G_h^1\rangle\rangle$, let V_i $(1 \le i \le h)$ be a set of vertices for G_i^0 and G_i^1 and let $b \in \{0, 1\}$. Here we define a function f_{UGIT} for UGIT as follows:

$$f_{\mathrm{UGIT}}(x, b, \langle \pi_1, \ldots, \pi_h\rangle) = (\pi_1(G_1^b), \ldots, \pi_h(G_h^b)),$$

where π_i is a random permutation on V_i $(1 \le i \le h)$.

Assume that $x \in$ UGIT. It follows from Definition 3.1 that $G_i^0 \simeq G_i^1$ for each i $(1 \le i \le h)$. Then the distribution $f_{\mathrm{UGIT}}(x, 0)$ over $\pi_1, \ldots, \pi_h$ is *identical* to the distribution $f_{\mathrm{UGIT}}(x, 1)$ over $\pi_1, \ldots, \pi_h$. Thus f_{UGIT} is positively opaque. Assume that $x \notin$ UGIT. It follows from Definition 3.1 that there exists an i_0 such that $G_{i_0}^0 \not\simeq G_{i_0}^1$. This implies that $\pi_{i_0}(G_{i_0}^0) \ne \varphi_{i_0}(G_{i_0}^1)$ for any permutations π_{i_0}, φ_{i_0} on V_{i_0}. Then

$$f_{\mathrm{UGIT}}(x, 0, \langle \pi_1, \ldots, \pi_h\rangle) \ne f_{\mathrm{UGIT}}(x, 1, \langle \varphi_1, \ldots, \varphi_h\rangle),$$

for any permutations π_i, φ_i on V_i. Thus f_{UGIT} is negatively transparent. □

For $h = 1$, the idea of Proposition 3.4 is inspired by existing protocols. This traces back to the protocol for graph isomorphism [13] to some extent but is more apparently influenced by the protocol for graph isomorphism [2] in which the bit commitment based on the graph isomorphism is fairly explicitly used. For every known random self-reducible language, e.g., quadratic residuosity, multiplicative subgroup $\langle g \rangle_p$ of Z_p^*, etc., we can define a language similar to UGIT and thus we can show in a way similar to Proposition 3.4 that such a language induces a positively opaque and negatively transparent function.

Proposition 3.5. EGIT *induces a positively opaque and negative transparent function.*

Proof. Let $x = \langle h, \langle G_1^0, G_1^1 \rangle, \langle G_2^0, G_2^1 \rangle, \ldots, \langle G_h^0, G_h^1 \rangle \rangle$, let V_i $(1 \le i \le h)$ be a set of vertices for G_i^0 and G_i^1, and let $b \in \{0, 1\}$. Here we define a function f_{EGIT} for EGIT as follows:

$$f_{\mathrm{EGIT}}(x, b, \langle\langle e_1, \ldots, e_h \rangle, \langle \pi_1, \ldots, \pi_h \rangle\rangle) = \left(b \oplus \left(\bigoplus_{i=1}^{h} e_i \right), \pi_1(G_1^{e_1}), \ldots, \pi_h(G_h^{e_h}) \right),$$

where $e_i \in \{0, 1\}$ is a random bit and π_i is a random permutation on V_i $(1 \le i \le h)$.

Assume that $x \in$ EGIT. It follows from Definition 3.2 that there exists an i_0 such that $G_{i_0}^0 \simeq G_{i_0}^1$. Then the distribution of random isomorphic copies of $G_{i_0}^0$ is identical to that of random isomorphic copies of $G_{i_0}^1$. This implies that the distribution $f_{\mathrm{EGIT}}(x, 0)$ over $e_1, \ldots, e_h, \pi_1, \ldots, \pi_h$ is *identical* to the distribution $f_{\mathrm{EGIT}}(x, 1)$ over $e_1, \ldots, e_h, \pi_1, \ldots, \pi_h$. Thus f_{EGIT} is positively opaque. Assume that $x \notin$ EGIT. It follows from Definition 3.2 that, for each i $(1 \le i \le h)$, $G_i^0 \not\simeq G_i^1$. Then, for any e_i, $d_i \in \{0, 1\}$ and any permutations π_i, φ_i on V_i,

$$f_{\mathrm{EGIT}}(x, 0, \langle\langle e_1, \ldots, e_h \rangle, \langle \pi_1, \ldots, \pi_h \rangle\rangle) \ne f_{\mathrm{EGIT}}(x, 1, \langle\langle d_1, \ldots, d_h \rangle, \langle \varphi_1, \ldots, \varphi_h \rangle\rangle).$$

Thus f_{EGIT} is negatively transparent. □

Again, for every known random self-reducible language, we can define a language similar to EGIT and thus we can show in a way similar to Proposition 3.5 that such a language induces a positively opaque and negatively transparent function.

Proposition 3.6. 1MOD4 *induces a positively opaque and negatively transparent function.*

Proof. Let $x = p_1^{e_1} p_2^{e_2} \cdots p_h^{e_h}$ be the prime factorization and let $b \in \{0, 1\}$. Here we define a function f_{1MOD4} for 1MOD4 as follows: $f_{\mathrm{1MOD4}}(x, b, r) = (-1)^b r^2 (\bmod x)$, where r is randomly chosen from Z_x^*. Note that -1 is a quadratic residue modulo x iff $x \in$ 1MOD4.

Assume that $x \in$ 1MOD4. From Definition 3.3 and the fact that -1 is a quadratic residue modulo x, it follows that, for any b and r, $f_{\mathrm{1MOD4}}(x, b, r)$ is a quadratic residue modulo x. This implies that the distribution $f_{\mathrm{1MOD4}}(x, 0)$ over $r \in Z_x^*$ is *identical* to the distribution $f_{\mathrm{1MOD4}}(x, 1)$ over $r \in Z_x^*$. Thus f_{1MOD4} is positively opaque. Assume that

$x \notin$ 1MOD4. From Definition 3.3 and the fact that -1 is a quadratic nonresidue modulo x, it follows that, for any $r \in Z_x^*$, $f_{1\text{MOD4}}(x, b, r) \equiv (-1)^b r^2 (\bmod x)$ is a quadratic residue modulo x iff $b = 0$. Then, for any $r, s \in Z_x^*$, $f_{1\text{MOD4}}(x, 0, r) \not\equiv f_{1\text{MOD4}}(x, 1, s)$. Thus $f_{1\text{MOD4}}$ is negatively transparent. □

It is not difficult to show that (1) $2 \in Z_N^*$ is a quadratic residue modulo N if and only if $N \in \pm 1$MOD8; (2) $3 \in Z_N^*$ is a quadratic residue modulo N if and only if $N \in \pm 1$MOD12; and (3) $5 \in Z_N^*$ is a quadratic residue modulo N if and only if $N \in \pm 1$MOD5. Then in a way similar to Proposition 3.6, we can show the following:

Proposition 3.7. ± 1MOD8, ± 1MOD12, *and* ± 1MOD5 *induce positively opaque and negatively transparent functions* $f_{\pm 1\text{MOD8}}$, $f_{\pm 1\text{MOD12}}$, *and* $f_{\pm 1\text{MOD5}}$, *respectively.*

4. Main Results

4.1. *Positively Opaque and Negatively Transparent Functions*

Assume that a language L induces a positively opaque and negatively transparent function f_L. Now we consider the following interactive protocol $\langle A, B \rangle$ for L: Let $x \in \{0, 1\}^*$ be a common input to $\langle A, B \rangle$. (A1) A randomly chooses $b \in \{0, 1\}$, $r \in \{0, 1\}^{k(|x|)}$, and sends $a = f_L(x, b, r)$ to B; (B1) B randomly chooses $e \in \{0, 1\}$ and sends e to A; (A2) A sends B $\sigma \in \{0, 1\}^{k(|x|)}$ such that $a = f_L(x, e, \sigma)$; and (B2) B checks that $a = f_L(x, e, \sigma)$. After $n = |x|$ independent invocations from step A1 to step B2, B accepts x iff every check in step B2 is successful.

From the fact that f_L is positively opaque and negatively transparent, ww can show the following in almost the same way as the case of random self-reducible languages [21].

Theorem 4.1. *If a language L induces a positively opaque and negatively transparent function, then there exists an unbounded round perfect zero-knowledge proof for L.*

As an immediate corollary to Theorem 4.1, we can show the following:

Corollary 4.2 (to Theorem 4.1). *Any $\mathcal{NP}$-complete language does not induce a positively opaque and negatively transparent function unless the polynomial hierarchy collapses.*

Proof. Fortnow [11] showed that if a language L has a statistical zero-knowledge proof, then $L \in$ co-$\mathcal{AM}$[1] and Boppana *et al.* [6] showed that if co-$\mathcal{NP} \subseteq \mathcal{AM}$, then the polynomial-time hierarchy collapses. The corollary follows from these and Theorem 4.1. □

---

[1] Goldreich *et al.* [15] pointed out that the proof of the result by Fortnow [11] has a flaw. Aiello and Håstad [1] contains a proof of that claim.

In the protocol $\langle A, B\rangle$, however, A needs to evaluate $\sigma \in \{0,1\}^{k(|x|)}$ such that $a = f_L(x, e, \sigma)$ for each iteration. Thus, in general, $\langle A, B\rangle$ could not be prover-practical. In this subsection we show a stronger result, i.e., L has a prover-practical perfect zero-knowledge proof. The protocol given below generalizes the protocol for graph isomorphism [13] and indeed coincides with it in the case of L being UGIT with $h = 1$.

Theorem 4.3. *If a language L induces a positively opaque and negatively transparent function, then there exists a prover-practical unbounded round perfect zero-knowledge proof for L.*

Proof. Since the language L induces a positively opaque and negatively transparent function f_L, $L \in \mathcal{NP}$ (see Definition 2.3). Let $x \in \{0,1\}^*$ be a common input to $\langle P, V\rangle$. Fix a polynomial-time computable function g_L that reduces L to the directed Hamiltonian cycle (DHAM), i.e., $x \in L$ iff $g_L(x) \in$ DHAM. Here we overview the outline of the interactive protocol $\langle P, V\rangle$ for L. P and V first reduce L to DHAM via the function g_L and then execute the zero-knowledge proof for DHAM [5] using (as a bit commitment) the positively opaque and negatively transparent function f_L. Recall that the prover uses a transparent bit commitment in the zero-knowledge proof for DHAM [5]. Then the transparent property of the bit commitment guarantees the soundness of the protocol, but the protocol is only computational (not perfect) zero-knowledge. For specificity, here we choose the zero-knowledge proof for DHAM but the ones for any other $\mathcal{NP}$-complete language would work.

Interactive Protocol $\langle P, V\rangle$ for L

common input: $x \in \{0,1\}^*$.

Initial: P and V reduces L to DHAM via the function g_L, i.e., $G = g_L(x)$. Let $A_G = (a_{ij})$ be the adjacency matrix of $G = (V, E)$ and let $n = |V|$.

P1-1: P randomly chooses $s_{ij} \in \{0,1\}^{k(|x|)}$ and a permutation π on V $(1 \le i, j \le n)$.

P1-2: P computes $c_{ij} = f_L(x, a_{\pi(i)\pi(j)}, s_{ij})$.

$P \to V$: $C = (c_{ij})$ $(1 \le i, j \le n)$.

V1: V randomly chooses $e \in \{0,1\}$.

$V \to P$: e.

P2-1: For $e = 0$, P assigns $\langle \pi, s_{11}, s_{12}, \ldots, s_{nn}\rangle$ to w.

P2-2: For $e = 1$, P assigns $\langle \langle i_1, j_1\rangle, \langle i_2, j_2\rangle, \ldots, \langle i_n, j_n\rangle, s_{i_1 j_1}, s_{i_2 j_2}, \ldots, s_{i_n j_n}\rangle$ to w such that $\langle i_1, j_1\rangle, \langle i_2, j_2\rangle, \ldots, \langle i_n, j_n\rangle$ is a single cycle.

$P \to V$: w.

V2-1: For $e = 0$, V checks that $c_{ij} = f_L(x, a_{\pi(i)\pi(j)}, s_{ij})$ for each i, j $(1 \le i, j \le n)$.

V2-2: For $e = 1$, V checks that $\langle i_1, j_1\rangle, \langle i_2, j_2\rangle, \ldots, \langle i_n, j_n\rangle$ is indeed a single cycle and that $c_{i_m j_m} = f_L(x, 1, s_{i_m j_m})$ for each m $(1 \le m \le n)$.

After n independent invocations from step P1-1 to step V2-2, V accepts x iff every check in step V2-1 and step V2-2 is successful.

In a way similar to the zero-knowledge proof for DHAM [5], we can show that the protocol $\langle P, V\rangle$ is a prover-practical perfect zero-knowledge proof for L. The completeness and prover-practicality are obvious. The soundness follows from the fact that f_L is negatively transparent. The perfect zero-knowledgeness follows from the fact that f_L is positively opaque. □

For a language $L \in \mathcal{NP}$, let ρ_L be the defining predicate of L. Define relation R_L to be $\langle x, y\rangle \in R_L$ iff $\rho_L(x, y) = 1$. Then we can show the following:

Corollary 4.4 (to Theorem 4.3). *If a language L induces a positively opaque and negatively transparent function, then there exists a perfect zero-knowledge proof of knowledge for* R_L.

Proof. This follows from the fact that the reduction from any $L \in \mathcal{NP}$ to DHAM is witness-preserving and polynomial-time invertible. □

4.2. *Positively Transparent and Negatively Opaque Functions*

Here we consider the case contrary to Theorem 4.3, i.e., the case that L induces a positively transparent and negatively opaque function (see Definition 2.3), and show that if a language L induces a positively transparent and negatively opaque function, then there exists a bounded round perfect zero-knowledge proof for L. The protocol given below generalizes a constant round perfect zero-knowledge proof for quadratic nonresiduosity [16], graph nonisomorphism [13], and the complement of random self-reducible languages [21].

Theorem 4.5. *If a language L induces a positively transparent and negatively opaque function, then there exists a two-round perfect zero-knowledge proof for L.*

Proof. Let L be a language that induces a positively transparent and negatively opaque function f_L. Let $x \in \{0, 1\}^*$ be a common input to $\langle P, V\rangle$. Here we overview the outline of the interactive protocol $\langle P, V\rangle$ for L. For each i $(1 \le i \le |x|)$, V randomly chooses $e_i \in \{0, 1\}$, $r_i \in \{0, 1\}^{k(|x|)}$, and computes $\alpha_i = f_L(x, e_i, r_i)$. Then V defines the following $\mathcal{NP}$-statement,

$$\exists e_1, e_2, \ldots, e_{|x|} \exists r_1, r_2, \ldots, r_{|x|} \quad \text{s.t.} \quad \bigwedge_{i=1}^{|x|} \alpha_i = f_L(x, e_i, r_i). \tag{1}$$

Fix a polynomial-time computable function g that reduces the $\mathcal{NP}$-statement of (1) to DHAM $G = (V, E)$, i.e., $G = g(\alpha_1, \ldots, \alpha_{|x|})$. Let H be a Hamiltonian cycle of G. From the witness-preserving property of the reduction from any $L \in \mathcal{NP}$ to DHAM, there exists a polynomial-time computable function h that satisfies

$$H = h(\langle \alpha_1, \ldots, \alpha_{|x|}\rangle, \langle e_1, \ldots, e_{|x|}; r_1, \ldots, r_{|x|}\rangle).$$

Then V generates polynomially many random copies isomorphic to G and commits to them with the positively transparent and negatively opaque function f_L. After these preliminary steps, V shows P that V knows the Hamiltonian cycle H of G. If V succeeds in convincing P, then P shows V that P knows $e_1, e_2, \ldots, e_{|x|}$.

The idea behind the protocol is almost the same as that of the perfect zero-knowledge proof for graph nonisomorphism [13]. Recall that the verifier uses a positively transparent and negatively opaque bit commitment in the perfect zero-knowledge proof for graph nonisomorphism [13]. Then the positively transparent and negatively opaque properties of the bit commitment guarantee the completeness and the soundness of the protocol, respectively. The perfect zero-knoweldgeness follows from the positively transparent property of the bit commitment.

Interactive Protocol $\langle P, V\rangle$ for L

common input: $x \in \{0, 1\}^*$.

V1-1: V randomly chooses $e_i \in \{0, 1\}$ and $r_i \in \{0, 1\}^{k(|x|)}$ for each i $(1 \le i \le |x|)$.
V1-2: V computes $\alpha_i = f_L(x, e_i, r_i)$.
V1-3: V computes $G = g(\alpha_1, \alpha_2, \ldots, \alpha_{|x|})$, i.e., V reduces the $\mathcal{NP}$-statement of (1) to DHAM $G = (V, E)$. Let $n = |V|$.
V1-4: V defines an adjacency matrix $A_G = (a_{ij})$ of G.
V1-5: V computes $H = h(\langle \alpha_1, \alpha_2, \ldots, \alpha_{|x|}\rangle, \langle e_1, e_2, \ldots, e_{|x|}; r_1, r_2, \ldots, r_{|x|}\rangle)$, where H is one of the Hamiltonian cycles of G.
V1-6: V randomly chooses a permutation π_ℓ on V $(1 \le \ell \le n^2)$ and $s_{ij}^\ell \in \{0, 1\}^{k(|x|)}$ $(1 \le i, j \le n)$.
V1-7: V computes $c_{ij}^\ell = f_L(x, a_{\pi_\ell(i)\pi_\ell(j)}, s_{ij}^\ell)$.
$V \to P$: $\langle \alpha_1, \alpha_2, \ldots, \alpha_{|x|}\rangle, \langle (c_{ij}^1), (c_{ij}^2), \ldots, (c_{ij}^{n^2})\rangle$ $(1 \le i, j \le n)$.
P1: P randomly chooses $b_\ell \in \{0, 1\}$ for each ℓ $(1 \le \ell \le n^2)$.
$P \to V$: $\langle b_1, b_2, \ldots, b_{n^2}\rangle$.
V2-1: If $b_\ell = 0$ $(1 \le \ell \le n^2)$, V assigns $\langle \pi_\ell, s_{11}^\ell, s_{12}^\ell, \ldots, s_{nn}^\ell\rangle$ to w_ℓ.
V2-2: If $b_\ell = 1$ $(1 \le \ell \le n^2)$, V assigns $\langle\langle i_1^\ell, j_1^\ell\rangle, \langle i_2^\ell, j_2^\ell\rangle, \ldots, \langle i_n^\ell, j_n^\ell\rangle, s_{i_1^\ell j_1^\ell}^\ell, s_{i_2^\ell j_2^\ell}^\ell, \ldots, s_{i_n^\ell j_n^\ell}^\ell\rangle$ to w_ℓ such that $\langle i_1^\ell, j_1^\ell\rangle, \langle i_2^\ell, j_2^\ell\rangle, \ldots, \langle i_n^\ell, j_n^\ell\rangle$ is a single cycle.
$V \to P$: $\langle w_1, w_2, \ldots, w_{n^2}\rangle$.
P2-1: P computes $G = g(\alpha_1, \alpha_2, \ldots, \alpha_{|x|})$ and an adjacency matrix $A_G = (a_{ij})$ of G.
P2-2: For each $b_\ell \doteq 0$ $(1 \le \ell \le n^2)$, if $c_{ij}^\ell = f_L(x, a_{\pi_\ell(i)\pi_\ell(j)}, s_{ij}^\ell)$ for each i, j $(1 \le i, j \le n)$, then P continues; otherwise P halts and rejects $x \in \{0, 1\}^*$.
P2-3: For each $b_\ell = 1$ $(1 \le \ell \le n^2)$, if $\langle i_1^\ell, j_1^\ell\rangle, \langle i_2^\ell, j_2^\ell\rangle, \ldots, \langle i_n^\ell, j_n^\ell\rangle$ is indeed a single cycle and $c_{i_m^\ell j_m^\ell}^\ell = f_L(x, 1, s_{i_m^\ell j_m^\ell}^\ell)$ for each m $(1 \le m \le n)$, then P continues; otherwise P halts and rejects x.
P2-4: If there exist $\beta_i \in \{0, 1\}$ and $t_i \in \{0, 1\}^{k(|x|)}$ such that $\alpha_i = f_L(x, \beta_i, t_i)$ for every i $(1 \le i \le |x|)$, then P continues; otherwise P halts and rejects x.
$P \to V$: $\langle \beta_1, \beta_2, \ldots, \beta_{|x|}\rangle$.
V3: If $\beta_i = e_i$ for every i $(1 \le i \le |x|)$, then V halts and accepts x; otherwise V halts and rejects x.

In a way similar to the perfect zero-knowledge proof for graph nonisomorphism [13], we can show that the protocol $\langle P, V\rangle$ is a two round perfect zero-knowledge proof for L. The completeness follows from the fact that f_L is positively transparent and the soundness follows from the fact that f_L is negatively opaque and the fact that V's proof of knowledge on the Hamiltonian cycle H of G is *perfectly* witness indistinguishable [10]. The perfect zero-knowledgeness follows from the fact that f_L is positively transparent and that fact that the reduction g from the $\mathcal{NP}$-statement of (1) to DHAM is polynomial-time invertible. □

5. Concluding Remarks

From Theorem 4.3, it follows that any language inducing a positively opaque and negatively transparent function has an unbounded round perfect zero-knowledge **Arthur–Merlin** proof. This could be improved, however, because, from Definition 2.3, any language inducing a positively opaque and negatively transparent function is in $\mathcal{NP}$, i.e., any language inducing a positively opaque and negatively transparent function has an $\mathcal{NP}$-proof [16]. Indeed, UGIT and EGIT are polynomial-time many–one reducible to graph isomorphism [9], [17], [18], and graph isomorphism has a bounded round perfect zero-knowledge proof [2]. Then:

1. If a language L induces a positively opaque and negatively transparent function, then does there exist a bounded round perfect zero-knowledge proof for L?

To affirmatively solve this question, a verifier will have to flip private coins, because Goldreich and Krawczyk [12] showed that there exists a bounded round zero-knowledge Arthur–Merlin proof for a language L, then $L \in \mathcal{BPP}$.

Every known random self-reducible language, e.g., graph isomorphism, quadratic residuosity, multiplicative subgroup $\langle g\rangle_p$ of Z_p^*, etc., induces a positively opaque and negatively transparent function, however, it is not known whether every random self-reducible language induces a positively opaque and negatively transparent function. Then:

2. Does every random self-reducible language induce a positively opaque and negatively transparent function?

Acknowledgments

The authors wish to thank the referees for their many helpful suggestions to improve the quality and the readability of the paper.

References

[1] Aiello, W., and Håstad, J., Statistical Zero-Knowledge Languages Can Be Recognized in Two Rounds, *J. Comput. System Sci.*, Vol. 42, No. 3, pp. 327–345 (1991).

[2] Bellare, M., Micali, S., and Ostrovsky, R., Perfect Zero-Knowledge in Constant Rounds, *Proceedings of the 22nd Annual ACM Symposium on the Theory of Computing*, pp. 482–493 (1990).

[3] Bellare, M., Micali, S., and Ostrovsky, R., The (True) Complexity of Statistical Zero-Knowledge, *Proceedings of the 22nd Annual ACM Symposium on the Theory of Computing*, pp. 494–502 (1990).

[4] Ben-Or, M., Goldreich, O., Goldwasser, S., Håstad, J., Kilian, J., Micali, S., and Rogaway, P., Everything Provable Is Provable in Zero-Knowledge, *Proceedings of Crypto* '88, Lecture Notes in Computer Science, Vol. 403, pp. 37–56 (1990).

[5] Blum, M., How To Prove a Theorem so No One Else Can Claim It, *Proceedings of the ICM*, pp. 1444–1451 (1986).

[6] Boppana, R., Håstad, J., and Zachos, S., Does co-$\mathcal{NP}$ Have Short Interactive Proofs?, *Inform. Process. Lett.*, Vol. 25, No. 2, pp. 127–132 (1987).

[7] Boyar, J., Friedl, K., and Lund, C., Practical Zero-Knowledge Proof: Giving Hints and Using Deficiencies, *J. Cryptology*, Vol. 4, No. 3, pp. 185–206 (1991).

[8] Brassard, G., Chaum, D., and Crépeau, C., Minimum Disclosure Proofs of Knowledge, *J. Comput. System Sci.*, Vol. 37, No. 2, pp. 156–189 (1988).

[9] Chang, R., On the Structure of Bounded Queries to Arbitrary $\mathcal{NP}$ Sets, *Proceedings of the 4th Structure in Complexity Theory Conference*, pp. 250–258 (1989).

[10] Feige, U., and Shamir, A., Zero-Knowledge Proofs of Knowledge in Two Rounds, *Proceedings of Crypto* '89, Lecture Notes in Computer Science, Vol. 435, pp. 526–544 (1990).

[11] Fortnow, L., The Complexity of Perfect Zero-Knowledge, *Proceedings of the 19th Annual ACM Symposium on Theory of Computing*, pp. 204–209 (1987).

[12] Goldreich, O., and Krawczyk, H., On the Composition of Zero-Knowledge Proof Systems, *Proceedings of ICALP* '90, Lecture Notes in Computer Science, Vol. 443, pp. 268–282 (1990).

[13] Goldreich, O., Micali, S., and Wigderson, A., Proofs that Yield Nothing but Their Validity or All Languages in $\mathcal{NP}$ have Zero-Knowledge Proof Systems, *J. Assoc. Comput. Mach.*, Vol. 38, No. 1, pp. 691–729 (1991).

[14] Goldreich, O., and Oren, Y., Definitions and Properties of Zero-Knowledge Proof Systems, *J. Cryptology*, Vol. 7, No. 1, pp. 1–32 (1994).

[15] Goldreich, O., Ostrovsky, R., and Petrank, E., Computational Complexity and Knowledge Complexity, *Proceedings of the 26th Annual ACM Symposium on the Theory of Computing*, pp. 534–543 (1994).

[16] Goldwasser, S., Micali, S., and Rackoff, C., The Knowledge Complexity of Interactive Proof Systems, *SIAM J. Comput.*, Vol. 18, No. 1, pp. 186–208 (1989).

[17] Köbler, J., Schöning, U., and Torán, J., *The Graph Isomorphism Problem: Its Structural Complexity*, Birkhäuser, Boston (1993).

[18] Lozano, A., and Torán, L., On the Nonuniform Complexity of the Graph Isomorphism Problem, *Proceedings of the 7th Structure in Complexity Theory Conference*, pp. 118–131 (1992).

[19] Naor, M., Ostrovksy, R., Venkatesan, R., and Yung, M., Perfect Zero-Knowledge Arguments for $\mathcal{NP}$ Can Be Based on General Complexity Assumptions, *Proceedings of Crypto* '92, Lecture Notes in Computer Science, Vol. 740, pp. 196–214 (1993).

[20] Ostrovsky, R., Venkatesan, R., and Yung, M., Secure Commitment Against a Powerful Adversary, *Proceedings of STACS* '92, Lecture Notes in Computer Science, Vol. 577, pp. 439–448 (1992).

[21] Tompa, M., and Woll, H., Random Self-Reducibility and Zero-Knowledge Interactive Proofs of Possession of Information, *Proceedings of the 28th Annual IEEE Symposium on Foundations of Computer Science*, pp. 472–482 (1987).