Perfectly Hiding Commitment Scheme with Two-Round from Any One-Way Functions *

Chunming Tang[1,†] Dingyi Pei[1] Zhuojun Liu[2] Zheng-an Yao[3] Mingsheng Wang[4]
1 School of Mathematics and Information Sciences, Guangzhou University, China(510006)
2 Key Laboratory of Mathematics Mechanization, AMSS, CAS, China(100080)
3 School of Mathematics and Statistics, Zhongshan University, China(510006)
4 State Key Laboratory of Information Security, Institute of Software, CAS China(100080)

Abstract

In FOCS2006, STOC2006, STOC2007, statistically hiding and computationally binding commitment schemes were constructed under the existence of one-way functions respectively, however, all of them have polynomial number of rounds complexity, i.e., impractical.

In this paper, we will firstly construct a perfectly hiding and computationally binding commitment scheme with two-round under the existence of one-way function. Comparing with all of previous commitments from any one-way function, our scheme has two excellent advantages: *perfectly hiding* and *two-round complexity*.

Keywords: Cryptography, Σ-protocol, perfectly hiding and computationally binding commitment scheme.

1 Background and Motivation

Commitment schemes are arguably among the most important and useful primitives in cryptography. Intuitively a commitment scheme is a two-party(interactive) protocol between a sender $\mathcal{P}$ and a receiver $\mathcal{V}$ in which after the sender $\mathcal{P}$ commits to a value b at hand, (1) the sender $\mathcal{P}$ cannot change his mind(this is known as the *binding* property); and (2) the receiver $\mathcal{V}$ learns nothing about the value of the bit b (this is known as the *hiding* property). Commitments have diverse applications to cryptographic protocols, such as zero-knowledge proofs, multi-party computation, digital auctions and electronic commerce[6, 9, 10, 11]. According to the computational power of senders and receivers, commitments can be classified into the four possible types shown in following Table[15].

Table 1. Classification of commitments.

Type	Computational power of sender P	Computational power of receiver V
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

Feige and Shamir[7] constructed a commitment of Type A under the existence of one-way functions, and Goldreich et al. [10] also constructed the commitment scheme of Type C from

*Partially supported by National Science Foundation of China (90604034(key project), 10726012) and 973 Project (2007CB311201)

†Email: tangcm622@hotmail.com

any one-way functions. In [22], Ostrovsky et al. showed that it is impossible to implement the commitment scheme of Type D.

The early construction of commitment schemes of Type B were based on specific number-theoretic complexity assumptions [1, 2], and were later generalized to any family of claw-free permutations [8], and then to any family of collision-resistant hash function [20]. In 1992, Naor et al. [18] showed that the collision resistance criterion is not necessary, by giving a beautiful construction of statistically hiding commitments and thus statistical zero-knowledge arguments for $\mathcal{NP}$ from any one-way permutation. They left as an open question whether these primitives could be based on arbitrary one-way functions, which could again be essentially minimal by [21, 23]. The progress in the past decade came in 2005 when Haitner et al. [13] showed how to construct statistically hiding commitments from any "approximable preimage size" one-way function, which is an one-way function where we can efficiently approximate the pre-image size of points in the range. Nguyen et al. [17, 19] in 2006 and Haitner et al. [12] in 2007 constructed Type B commitment from any one-way function respectively, however, their schemes have polynomial number of rounds.

In this paper, we will devote to look for a way to construct a perfectly hiding and computationally binding commitment scheme, which is a commitment scheme of Type B, under the existence of one-way function. We will make use of Σ-protocol as a main tool.

Σ-protocol is a three-move interactive protocol between the prover and the verifier which the verifier is only required to send random bits as a challenge to the prover. Σ-protocol has become an important cryptographic primitive because of its following excellent characters: 1)*validity*; 2) *special soundness*; 3) *honest-verifier zero-knowledge*. The term Σ-protocol was introduced by Cramer for the reason that he called these protocols Σ-protocols is that the shape of the letter Σ[4]. So far, lots of cryptographic protocols have been constructed based on Σ-protocols, such as, identification schemes, digital signature schemes, secret sharing schemes[4]. Comparing with traditional cryptographic protocols, these protocols based on Σ-protocols have better security and more applicability.

Based on Σ-protocol, a new method to construct a commitment scheme was proposed in [14], furthermore, Damgard proved that these commitment are perfectly hiding. However, we find that these commitment scheme based on Σ-protocols are perfectly hiding only when Σ-protocols satisfy some special conditions. In addition, we obtain the most important result according to the method in [14]: *there exists perfectly hiding commitments if one-way function exists, furthermore, these commitments only need two rounds.*

We emphasis that we firstly construct perfectly hiding commitments if one-way function exists, however, only statistically hiding commitments, which is weaker than perfectly hiding commitments, can be constructed from any one-way function [12, 17, 19]. Especially, our commitment schemes are more efficient than any previous commitment for two-round complexity of our commitments.

1.1 Our Contribution

1) We will construct two perfectly hiding and computationally binding commitment schemes from Σ-protocol with computational zero-knowledge and perfect zero-knowledge, respectively. From these commitment schemes, we improve the result in [14], and obtain new result showed in theorem 3 in this paper.
2) We will firstly construct a perfectly hiding and computationally binding commitment scheme under the existence of one-way function, furthermore, this commitment scheme only needs two rounds. Comparing with previous commitments, our commitment has two excellent advantages: *perfectly hiding* and *two-round complexity.*

1.2 Organization

We start with some basic definitions and properties on Σ-protocol and commitment scheme. Then, we introduce two perfectly hiding commitments based on Σ-protocol in section 3, and construct a perfectly hiding commitment from any one-way function in section 4.

2 Preliminaries

NP relations We say that a binary relation $R \subseteq \{0,1\}^* \times \{0,1\}^*$ is an **NP** relation if there exists a polynomial $p(\cdot)$ such that for any $(x,w) \in R, |w| \leq p(|x|)$ and in addition there exists a polynomial time Turing machine for deciding membership in R. We denote by L_R the following language: $L_R = \{x | \exists w \ s.t. \ (x,w) \in R\}$. We say that $L \in NP$ if $L = L_R$ for some NP relation R.
A negligible function is a function that grows slower that inverse of any polynomial. That is, $\nu : \mathbb{N} \to \mathbb{N}$ is negligible if for any positive polynomial $p(\cdot)$ there exists a number n_0 such that $\nu(n) < \frac{1}{p(n)}$ for all $n > n_0$. We will sometimes use $negl(\cdot)$ to denote some unspecified negligible function.
One-Way Function A function $f : \{0,1\}^* \to \{0,1\}^*$ is called *one-way* if the following conditions hold:
1. there exists a deterministic polynomial-time algorithm $\mathcal{A}$ such that on input x, $\mathcal{A}$ outputs $f(x)$;
2. for every non-uniform probabilistic polynomial-time algorithm $\mathcal{A}'$ there exists a negligible function ν such that for all sufficiently large k, it holds that

$$Prob(x \leftarrow \{0,1\}^*; \mathcal{A}'(f(x)) \in f^{-1}(f(x))) < \nu(k).$$

We call each sending of a message by a party a move, and say **a round** is two consecutive moves.

2.1 Commitment scheme

Commitment scheme is a basic building block and has diverse applications to cryptographic protocols, especially to zero-knowledge proofs[10, 15]. Informally, a commitment scheme is a two-stage protocol between a sender and a receiver. In the first stage, the sender commits to a value b, and in the second, the sender 'reveal' this value to the receiver. We want two security properties from a commitment scheme. The *hiding* property says that the receiver does not learn anything about the value b during the commit stage. And the *binding* property says that after the commit stage, there is at most one value that the sender can successfully open.

Definition 1 *(Gen, Com, Ver) is a* **commitment scheme***[6] if:*
- efficiency: *Gen, Com and Ver are polynomial-time algorithms;*
- completeness: *for all m it holds that*

$$Prob(crs \leftarrow Gen(1^k); (com, dec) \leftarrow Com(crs, m) : Ver(crs, com, dec, m) = 1) = 1$$

- binding: *for any polynomial-time algorithm sender there is a negligible function ν such that for all sufficiently large k it holds that*

$$Prob(crs \leftarrow Gen(1^k); (com, m_0, m_1, dec_0, dec_1) \leftarrow sender(crs) :$$

$$m_0 \neq m_1 \ and \ Ver(crs, com, dec_0, m_0) = Ver(crs, com, dec_1, m_1) = 1) \leq \nu(k)$$

- hiding: *for any adversary receiver there is a negligible function ν such that for all m_0, m_1 where $|m_0| = |m_1|$ and all sufficiently large k it holds that*

$$Prob(crs \leftarrow Gen(1^k); b \leftarrow \{0,1\}; (com, dec) \leftarrow Crs(crs, m_b) : b \leftarrow receiver(com)) < \frac{1}{2} + \nu(k)$$

A commitment is *statistically hiding* if for any computationally unbounded adversary receiver there is a negligible function ν such that for all m_0, m_1 where $|m_0| = |m_1|$ and all sufficiently large k it holds that

$$Prob(crs \leftarrow Gen(1^k); b \leftarrow \{0,1\}; (com, dec) \leftarrow Crs(crs, m_b) : b \leftarrow receiver(com)) < \frac{1}{2} + \nu(k)$$

And a commitment is *perfectly hiding* if for any computationally unbounded adversary receiver for all m_0, m_1 where $|m_0| = |m_1|$ and all sufficiently large k it holds that

$$Prob(crs \leftarrow Gen(1^k); b \leftarrow \{0,1\}; (com, dec) \leftarrow Crs(crs, m_b) : b \leftarrow receiver(com)) = \frac{1}{2}$$

Similarly, we can define *statistically binding* and *perfectly binding* commitments.

A commitment is *statistically binding* for any computationally unbounded sender there is a negligible function ν such that for all sufficiently large k it holds that

$$Prob(crs \leftarrow Gen(1^k); (com, m_0, m_1, dec_0, dec_1) \leftarrow sender(crs) :$$

$$m_0 \neq m_1 \ and \ Ver(crs, com, dec_0, m_0) = Ver(crs, com, dec_1, m_1) = 1) \leq \nu(k)$$

And a commitment is *perfectly binding* for any computationally unbounded sender for all sufficiently large k it holds that

$$Prob(crs \leftarrow Gen(1^k); (com, m_0, m_1, dec_0, dec_1) \leftarrow sender(crs) :$$

$$m_0 \neq m_1 \ and \ Ver(crs, com, dec_0, m_0) = Ver(crs, com, dec_1, m_1) = 1) = 0$$

Usually, if the binding property holds with respect to a computationally unbounded algorithm sender, the commitment scheme is said to be *unconditionally binding*; if instead, the hiding property holds with respect to a computationally unbounded algorithm receiver, the commitment scheme is said to be *unconditionally hiding*.

Hence "unconditionally binding (or hiding)" means "perfectly binding (or hiding)" or "statistically binding (or hiding)". Obviously, "perfectly binding (or hiding)" commitments must be "statistically binding (or hiding)" according to their definitions.

According to the computational power of senders and receivers, commitment scheme can be classified into the four possible types[15]: *computationally hiding and computationally binding* (Type A), *unconditionally hiding and computationally binding*(Type B), *computational hiding and unconditionally binding* (Type C), and *unconditionally hiding and unconditionally binding*(Type D).

For 4 types commitments, there have existed the following excellent results:

1: The commitment scheme of Type A exists if one-way function exist [7];

2: The commitment scheme of Type B exists if one-way functions exist[12, 17, 19], however, these commitments have polynomial number of rounds.

3: The commitment scheme of Type C exists if one-way functions exist [8];

4: It is impossible to construct commitment scheme of Type D [22].

In this paper, we mainly consider the construction of commitment scheme of Type B. We will construct perfectly hiding commitment schemes from any one-way function, furthermore, they have only two rounds complexity.

2.2 Σ-protocol

Let protocol (P, V) be a three-move interactive protocol between a prover P and a verifier V, where the prover acts first. The verifier is only required to send random bits as a challenge to the prover. For some $(x, w) \in R$, the common input to both players is x while w is private input to the prover. For such given x, let (a, e, z) denote the conversation between the prover and the verifier. To compute the first and final messages, the prover invokes efficient algorithms $a(\cdot)$ and

$z(\cdot)$, respectively, using (x, w) and random bits as input. Using an efficient predicate $\phi(\cdot)$, the verifier decides whether the conversation is acceptable with respect to x. The relation R, the algorithm $a(\cdot)$, $z(\cdot)$ and $\phi(\cdot)$ are public.

Definition 2 *The above protocol is said to be a* **Σ-protocol for relation** R *if it has the following properties:*

1. **(Validity)** *If P, V follow the protocol, the verifier always accepts.*
2. **(Special soundness)** *From any x and any pair of accepting conversations on input x, (a, e, z), (a, e', z') where $e \neq e'$, one can efficient compute w such that $(x, w) \in R$.*
3. **(Honest-verifier zero-knowledge)** *There exists a polynomial simulator M, which on input x and a random e outputs an accepting conversation of the form (a, e, z), with the same probability distribution as conversations between the honest P, V on input x.*

Proposition 1 *[14] The properties of Σ-protocols are invariant under parallel composition.*

3 Perfectly Hiding Commitment Scheme Based on Σ-protocol

Assume we are given a hard relation R with generator $\mathcal{G}$ and Σ-protocol $\mathcal{P}$. Assume also that it is easy to check membership in L_R, that is, given x, it is easy to decide if there exists w such that $(x, w) \in R$.

If Σ-protocol $\mathcal{P}$ is efficient, we can set up the following commitment scheme:

Commitment Scheme Based on $\Sigma -$ protocol

1. **(Set-up)** The receiver V runs (in private) generator $\mathcal{G}$ on input 1^k to get $(x, w) \in R$, sends x to P who checks that $x \in L_R$.
2. **(Commit)** To commit to a bit string e, P runs the simulator M on input x, e to get (a, e, z), and sends a to V.
3. **(Decommit)** To open the commitment, P sends e, z to V, who checks that (a, e, z) is an accepting conversation.

In order to understand the above scheme is computationally hiding or statistically hiding or perfectly hiding, we show following commitment schemes based on Σ-protocols.

Example 1:

Let p be a prime, q a prime divisor in $p - 1$, and g an element of order q in Z_p^*. Suppose a prover P has chosen w in Z_q at random and has published $h = g^w \bmod p$. A verifier V who gets p, q, g, h can check that p, q are prime, and that g, h have order q. Since there is only one subgroup of order q in Z_p^*, this automatically means that $h \in < g >$, i.e., there exists w such that $h = g^w$. But this does not necessarily mean that P knows such a w.

There exists a Σ-protocol which suggested by Schnorr [5] gives a very efficient way to convince V about this:

1 P chooses r at random in Z_q and sends $a = g^r \bmod p$ to V.

2 V chooses a challenge e at random in Z_q and sends it to P.

3 P sends $z = r + ew \bmod q$ to V.

4 V accepts the case that P holds a valid w if $g^z = ah^e \bmod p$, else, rejects.

We will construct commitment scheme based on the above Σ-protocol.

Commitment Scheme 1

1. **(Set-up)** The receiver V randomly chooses two primes p, q where q is a divisor in $p-1$, then chooses w in Z_q at random and computes $h = g^w \ mod \ p$. V sends (p, q, g, h) to the sender P who checks that p, q are prime and g, h have order q.

2. **(Commit)** To commit to a bit string e, P runs the simulator M on input e, p, q, g, h according to the following steps:

   (a) P chooses randomly z in Z_q.

   (b) P computes $a = \frac{g^z}{h^e}$.

   (c) P obtains a conversation (a, e, z) such that $g^z = ah^e \ mod \ p$

   P sends commitment a to string e to V.

3. **(Decommit)** To open the commitment, P sends e, z to V, who checks that (a, e, z) is an accepting conversation.

Theorem 1 *The commitment scheme 1 is perfectly hiding and computationally binding.*

Proof(Sketch): Efficiency and **Completeness** are obvious.

(Computationally Binding) If a polynomial-time bounded prover P^* could efficiently output a, and open it both as e, z and as e', z' with $e \neq e'$, then we would have accepting conversations in $\mathcal{P}$, (a, e, z), (a', e', z'), this means by definition 2 that we can compute w such that $h = g^w \ mod \ p$ efficiently, and this contradicts the assumption that discrete logarithm problem (DLP) is a hard problem.

(Perfectly Hiding) For any computationally unbounded receiver V, V can compute $r = \log_g a$ (also knows $w = \log_g h$). In order to obtain the committed value e from (r, w), V has to find a pair (e, z) such that $z = r + ew \ mod \ q$, i.e., $g^z = ah^e \ mod \ p$. Obviously, V can correctly guess e with probability $\frac{1}{q}$ for the reason that there exists a z' such that $z' = r + e'w \ mod \ q$ for any $e' \in Z_q$. Hence, it is perfectly hiding. ■

In example 1, Σ-protocol on DLP is computational zero-knowledge for honesty verifier. Assume that the honesty verifier has computationally unbounded power, he can compute w such that $h = g^w \ mod \ p$. Hence, *perfectly hiding commitment scheme can be constructed from Σ-protocol with computational zero-knowledge.*

Example 2:

Let p and q be two large primes such that q divides $p-1$, G_q is the unique subgroup of Z_p^* of order q, and g is a generator of G_q. Let h be another generator of G_q such that nobody knows $\log_g h$.

Assume P hold w_1 and w_2 such that $y = g^{w_1}h^{w_2} \ mod \ p$. There exists a Σ-protocol gives a very efficient way to convince V about this[3]:

1 P chooses r_1, r_2 at random in Z_q and sends $a = g^{r_1}h^{r_2} \ mod \ p$ to V.

2 V chooses a challenge e at random in Z_q and sends it to P.

3 P sends $z = (z_1, z_2)$ to V, where $z_i = r_i + ew_i \ mod \ q (i = 1, 2)$.

4 V accepts the case that P holds a valid w if $g^{z_1}h^{z_2} = ay^e \ mod \ p$, else, rejects.

A commitment scheme can be constructed based on the above Σ-protocol.

Commitment Scheme 2

1. **(Set-up)** The receiver V randomly chooses two primes p, q where q is a divisor in $p-1$, then chooses w_1, w_2 in Z_q at random and computes $y = g^{w_1}h^{w_2} \ mod \ p$, where g, h are generators of subgroup G_q in Z_p^*. V sends (p, q, g, h, y) to the sender P who checks that p, q are prime and g, h have order q.

2. **(Commit)** To commit to a bit string e, P runs the simulator M on input e, p, q, g, h, y according to the following steps:

(a) P chooses randomly $z = (z_1, z_2)$, where z_1, z_2 in Z_q.

(b) P computes $a = \frac{g^{z_1} h^{z_2}}{y^e}$.

(c) P obtains a conversation (a, e, z) such that $g^{z_1} h^{z_2} = a y^e \ mod\ p$

P sends commitment a to string e to V.

3. **(Decommit)** To open the commitment, P sends e, z to V, who checks that (a, e, z) is an accepting conversation.

Theorem 2 *The commitment scheme 2 is perfectly hiding and computationally binding.*

Proof(Sketch): Efficiency and **Completeness** are obvious.

(Computationally Binding) If a polynomial-time bounded prover P^* could efficiently output a, and open it both as e, z and as e', z' with $e \neq e'$, then we would have accepting conversations in $\mathcal{P}$, (a, e, z), (a', e', z'), this means by definition 2 that we can compute $w = (w_1, w_2)$ such that $y = g^{w_1} h^{w_2} \ mod\ p$ efficiently, and this contradicts the assumption that DLP is a hard problem.

(Perfectly Hiding) For any computationally unbounded receiver V, V can compute $w = \log_g h$, $r = r_1 + w r_2 = \log_g a$, $w' = w_1 + w w_2 = \log_g y$. In order to obtain the committed value e from (r, w, w'), V has to find a pair (e, z_1, z_2) such that $z_1 + w z_2 = r + w' e \ mod\ q$ from $g^{z_1} h^{z_2} = a y^e \ mod\ p$. Obviously, V can correctly guess e with probability $\frac{1}{q}$ for the reason that there exists pairs (z_1', z_2') such that $z_1' + w z_2' = r + w' e' \ mod\ q$ for any $e' \in Z_q$. Hence, it is perfectly hiding. ■

In example 2, Σ-protocol is perfect zero-knowledge for honesty verifier for the following reasons:

Assume the honesty verifier has computationally unbounded power,

1) he cannot compute w_1, w_2 such that $y = g^{w_1} h^{w_2} \ mod\ p$ even that he can compute $w = \log_g h$ and $w_1 + w w_2 (= \log_g y)$;

2) he cannot also obtain w_1, w_2 such that $y = g^{w_1} h^{w_2} \ mod\ p$ from a valid conversation (a, e, z) because cannot correctly guess w_1, w_2 from $w = \log_g h$, $r = r_1 + w r_2 = \log_g a$, $w' = w_1 + w w_2 = \log_g y$, $z_1 + w z_2 = r + w' e \ mod\ q$.

Hence, *perfectly hiding commitment scheme can also be constructed from Σ-protocol with perfect zero-knowledge.*

Now, we obtain the fact that perfectly hiding commitment scheme can be constructed from Σ-protocol with computational zero-knowledge or perfect zero-knowledge (including statistically zero-knowledge). Then, can we obtain this conclusion that perfectly hiding commitment scheme can be constructed from any Σ-protocol? Our answer are negative, but we can prove the following theorem.

Theorem 3 *The commitment scheme based on Σ-protocol will be a perfectly hiding commitment scheme with computational binding if the first message a is independent of the challenge e in Σ-protocol $\mathcal{P}$.*

Proof(Sketch):(perfectly Hiding) In a real life, P's first message a is independent of the challenge e. Since $x \in L_R$, simulation by M is perfect by definition of Σ-protocols, hence the a generated by M is uncorrelated to e. In other words, the first message a can construct a valid conversation (a, e, z) for any e if someone knows w such that $(x, w) \in R$. The computationally unbounded receiver V cannot obtain any information on the committed value e because he knows w such that $(x, w) \in R$, so V only guesses e at random.

(Computationally Binding) If a cheat prover P^* could efficiently output a, and open it both as e, z and as e', z' with $e \neq e'$, then we would have accepting conversations in $\mathcal{P}$, (a, e, z), (a', e', z'), this means by definition that we can compute w efficiently, and this contradicts the assumption that R is a hard problem. ■

From the above theorem, we can obtain perfectly hiding commitment schemes with computational binding. However, these commitment schemes are not constructed under the weakest

assumption that one-way function exist, for example, both of commitment scheme 1 and 2 are based on DLP. In the following section, we will construct perfectly hiding commitment scheme with computational binding from any one-way function.

4 Perfectly Hiding Commitment Scheme from Any One-Way Function

If we can construct a Σ-protocol from any one-way function, furthermore, the first message a of this Σ-protocol is independent of the challenge e, then we will be able to construct a perfectly hiding commitment scheme with computational binding from any one-way function according to theorem 3.

We recall computational zero-knowledge proof on Hamiltonian Cycle(in short HC) which is a NP-complete statement, which is revised from the protocol in [10] and [16].

Protocol HC

1. *Common input*: $G = (V', E)$, with $n = |V'|$.
2. *Auxiliary input to prover:* a directed Hamiltonian cycle, $C \subset E$, in G.
3. Step $P1$: The prover P selects a random permutation π of the vertices and commits to the entries of the adjacency matrix of the resulting permuted graph, sends these commitments to the verifier V. That is, he sends an n-by-n matrix of commitments such that the $(\pi(i), \pi(j))$ entry is a commitment to 1 if $(i, j) \in E$ and is a commitment to 0 otherwise.
4. Step $V1$: V uniformly selects $\sigma \in \{0, 1\}$ and sends it to P.
5. Step $P2$: If $\sigma = 0$, then P sends π to V along with the revealing of all commitments. Otherwise, P reveals to V only commitments to entries $(\pi(i), \pi(j))$, with $(i, j) \in C$.
6. Step $V2$: If $\sigma = 0$, then V checks that the revealed graph is indeed isomorphic, via π, to G. Otherwise, V simply checks that all revealed values are 1 and that the corresponding entries form a simple n-cycle. V accepts if and only if the corresponding condition holds.

Remark: *In Protocol HC, the prover P makes use of a commitment scheme of Type A or Type C which exists if one-way function exist.*

Theorem 4 *Protocol HC is a Σ-protocol if one-way function exists.*

Proof: 1) If P and V follow the protocol, the verifier always accepts. 2) Assume that the messages is denoted as a in step P1, the messages is denoted as e in step V1, and the messages is denoted as z in step P2. For a, one can compute a Hamiltonian Cycle for the graph G if he receives two conversations (a, e, z) and (a, e', z'). 3) If the verifier is honest, i.e., the challenge is random, then there exists a polynomial-time simulator M which can simulate the conversations between P and honest V by rewinding V, where the construction of the simulator can referred to [10].

Obviously, the protocol will be a Σ-protocol basing on 1), 2) and 3) if commitment scheme of Type A or Type C exist. It is well-known that commitment scheme of Type A and Type C exist if one-way function exist from [8, 10]. Hence, the above theorem follows.■

Protocol HC is also computational zero-knowledge proof of knowledge [10] for the reason that any computationally unbounded verifier V can extract information on HC from commitment in message a.

Now, we use Protocol HC to construct a bit commitment scheme, which is also referred in [7]:

Commitment Scheme 3

1. **(Set-up)** The receiver V runs (in private) generator $\mathcal{G}$ on input 1^k to get $G = (V', E)$ and its Hamiltonian Cycle C, sends G to P who verifies that $G \in L_R$.

2. **(Commit)** To commit to a bit $e \in \{0, 1\}$, P runs the simulator M on input G, e to get (a, e, z), and sends a to V. P obtains (a, e, z) by the following steps:

   (a) P commits to 0 by choosing a random permutation π, permuting the nodes of G, and committing to the entries of the resulting adjacency matrix. P may reveal the committed bit $'0'$ by revealing π and the entries of the matrix. That is, z is composed of π and the entries of the matrix.

   (b) P commits to 1 by choosing the n nodes clique and committing to its adjacency matrix (which is all 1). P may reveal the committed bit '1' by opening a random cycle in this matrix. That is, z is a random cycle in this matrix.

3. **(Decommit)** To open the commitment, P sends e, z to V, who checks that (a, e, z) is an accepting conversation.

Theorem 5 *The above bit commitment scheme is only computationally hiding and computationally binding.*

Proof(Sketch): Efficiency and **Completeness** are obvious.

(Computationally Binding) If a polynomial-time bounded prover P^* could efficiently output a, and open it both as $0, z$ and as $1, z'$, then we would have accepting conversations $(a, 0, z)$, $(a, 1, z')$, this means by definition of Σ-protocol that the P^* can compute a Hamiltonian Cycle of grape G, and this contradicts the assumption that HC is a NP-complete problem.

(Computationally Hiding) For any computationally unbounded receiver V, V can open commitment scheme used in commitment a because this commitment only is computationally hiding. After opening commitment a, V can decide the committed value $e = 0$ or $e = 1$ by the opened value because the revealed graph is isomorphic to G if $e = 0$, and all revealed values are 1 if $e = 1$. ■

In order to obtain perfectly hiding commitment scheme, we need hide the commitment a in case that any computationally unbounded receiver V can direct open it and obtain the committed value e. We improve the commitment scheme 3 and obtain the following commitment scheme.

Commitment Scheme 4

1. **(Set-up)** The receiver V runs (in private) generator $\mathcal{G}$ on input 1^k to get $G = (V', E)$ and its Hamiltonian Cycle C, sends G to P who verifies that $G \in L_R$.

2. **(Commit)** To commit to a bit $e \in \{0, 1\}$

   (a) P runs the simulator M on input G, e to get (a', e, z). The simulating process is similar in Commit phase in Commitment Scheme 3.

   (b) P runs a random number generator $\mathcal{G}'$ with input 1^k to get a random number r.

   (c) P computes $a = a' \oplus f(r)$ and send it to V, where $f : \{0, 1\}^* \to \{0, 1\}^{|a'|}$ is one-way function.

3. **(Decommit)** To open the commitment, P sends r, a', e, z to V, who checks that (a', e, z) is an accepting conversation and verifies $a = a' \oplus f(r)$.

Theorem 6 *The above commitment scheme is perfectly hiding and computationally binding.*

Proof(Sketch): Efficiency and **Completeness** are obvious.

(Computationally Binding) Assume a polynomial-time bounded prover P^* could efficiently output a, then P^* maybe do as followed:

(1): He opens the commitment a both as $r, a', 0, z$ and as $r, a', 1, z'$, then we would have accepting conversations $(a', 0, z)$, $(a', 1, z')$.

(2): He opens the commitment a both as $r, a', 0, z$ and as $r', a'', 1, z'$, then we would have accepting conversations $(a', 0, z)$, $(a'', 1, z')$, and $a' \oplus f(r) = a'' \oplus f(r')$.

In (1), it means by definition of Σ-protocol that the P^* can compute a Hamiltonian Cycle of grape G, and this contradicts the assumption that HC is a NP-complete problem. In (2), $(a', 0, z)$ and $(a'', 1, z')$ is easily simulated, however it is impossible to obtain r' such that $a' \oplus f(r) = a'' \oplus f(r')$ unless P^* can invert one-way function f.

(Perfectly Hiding) For any computationally unbounded receiver V, V cannot obtain any information about the committed value e from commitment a. Because V only knows the commitment a, however, a can be opened as 0 or 1 with probability $\frac{1}{2}$ if any one has a HC C of graph G. Although any one can obtain e from a', V can not obtain any information about a' from $a = a' \oplus f(r)$ because lots of pairs (a', r) such that $a = a' \oplus f(r)$ and every pair (a', r) has same probability. ■

Theorem 7 *Perfectly hiding and computationally binding commitment scheme exists if one-way function exist.*

Proof(Sketch): *Commitment Scheme 4* is perfectly hiding and computationally binding from theorem 6. Commitment scheme used in the first message a in Σ-protocol on HC is commitment scheme of Type A or Type C. The random number generator $\mathcal{G}'$ and commitment scheme of Type A or Type C exist if one-way function exists. Hence, we obtain this conclusion that perfectly hiding and computationally binding commitment scheme exists if one-way function exist. ■

In *Commitment Scheme 4*, the prover P must convince that graph G generated by the verifier V has Hamiltonian Cycle C by using a witness hiding protocol [7], that is, V has to prove that G has Hamiltonian Cycle C by using *Protocol HC*, which is proven to be a witness hiding protocol [24], before P runs the simulator. Hence, the number of round is two rounds in *Commit* stage in *Commitment Scheme 4*. The commitment schemes 1-4 can be composed parallel from Proposition 1.

Theorem 8 *There exist perfectly hiding and computationally binding commitment scheme with two rounds complexity if one-way function exist.*

Our commitment schemes have only one round complexity in the random oracle model[14].

References

[1] G. Brassard, D. Chaum, and C. Crepeau. *Minimum Disclosure Proofs of Knowledge.* J. Comput. Syst. Sci., 37(2): 156-189, 1988.

[2] J. Boyar, S. A. Kurtz, and M. W. Krentel. *A Discrete Logarithm Implementation of Perfect Zero-Knowledge Blobs.* J. Cryptology, 2(2): 63-76, 1990

[3] . D. Chaum, J.H. Evertse, and J van de Graaf. *An improved protocol for demonstrating possession of discrete logarithms and some generalizations.* Advances in Cryptology-EUROCRYPT'87, (1988)127-141

[4] R. Cramer.*Modular Design of Secure yet Practical Cryptographic Protocol.* PhD thesis, University of Amsterdam, 1997

[5] C. Schnorr. *Efficient Signature Generation by Smart Cards.* Journal of Cryptology. Vol. 4(3), 1991

[6] D. Catalano, and I. Visconti. *Hybrid Commitments and Their Applications to Zero-knowledge Proof Systems.* Theoretical Computer Science, 374(2007), 229-260, 2007

[7] U. Feige and A. Shamir. *Zero-knowledge proofs of knowledge in two rounds.* In CRYPTO'89, 526-545.

[8] O. Goldreich, and A. Kahan. *How to construct constant-round zero-knowledge proof systems for NP.* Journal of Cryptography, Vol 9, No.2, pp. 167-189, 1996

[9] S. Goldwasser, S. Micali, and C. Rackoff. *The knowledge Complexity of Interactive Proof Systems.* SIAM Journal on Computing, Vol. 18, pp. 186-208, 1989.

[10] O. Goldreich, *Foundations of Cryptography (Basic Tools)*, Cambridge University Press, 2001.

[11] O. Goldreich. *Secure Multi-Party Computation.* http://www.wisdom.weizmann.ac.il/ oded/pp.html

[12] I. Haitner, and O. Reingold. *Statistically-Hiding Commitment from Any One-Way Function.* In 39th STOC2007, pp. 1-10, 2007

[13] I. Haitner, O. Horvitz, J. Katz, C. Y. Koo, R. Morselli, and R. Shaltiel. *Reducing Complexity Assumptions for Statistically-Hiding Commitment.* In Proc. EUROCRYPT, pages 58-77, 2005.

[14] I. Damgard, *On Σ-protocols.* CPT 2004. Available at http://www.daimi.au.dk/ ivan/Sigma.ps, 2002

[15] T. Itoh, Y. Ohta, and H. Shizuya *A Language-Dependent Cryptographic Primitive.* J. Cryptology, 10(1): 37-49, 1997

[16] M. Blum. *How to prove a theorem so no one else can claim it.* In Proceedings of the International Congress of Mathematicians. Vol. 1,2 pp 1444-1451, 1987

[17] M. Nguyen, S. J. Ong, and S. Vadhan. *Statistical Zero-Knowledge Arguments for NP from Any One-Way Function.* In Proc. 47th FOCS, 2006

[18] M. Naor, R. Ostrovsky, R. Venkatesan, and M. Yung. *Perfect Zero-Knowledge Arguments for NP Using Any One-Way Permutation.* J. Cryptology, 11(2): 87-108, 1998.

[19] M. Nguyen, and S. Vadhan. *Zero-Knowledge with Efficient Provers.* In Proc. 38th STOC, pages 287-295, 2006

[20] M. Naor, and M. Yung. *Universal One-Way Hash Function and their Cryptographic Applications.* In Proc. 21st STOC, pages 33-43, 1989.

[21] R. Ostrovsky. *One-Way Functions, Hard on Average Problems, and Statistical Zero-Knowledge Proofs.* In Proc. 6th SICTC, pages 133-138, 1991

[22] R. Ostrovsky, R. Venkatesan, and M. Yung. *Secure Commitment Against a Powerful Adversary.* In Proc of STACS'92, LNCS577, pages 439-448, 1992

[23] R. Ostrovsky, and A. Wigderson. *One-Way Functions are Essential for Non-Trivial Zero-Knowledge.* In Proc. 2nd Israel Symposium on theory of computing systems, pages 3-17, 1993

[24] C.M Tang, D.Y Pei, Z.A Yao. *Efficient Zaps and Signatures of Knowledges.* In Proceeding of IEEE International Conference on Computational Intelligence and Security(CIS'2007), pages 637-641, 2007