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

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

Commitment schemes are arguably among the most important and useful primitives in cryptography. According to the computational power of receivers, commitments can be classified into three possible types: *computational hiding commitments, statistically hiding commitments* and *perfect computational commitments*. The fist commitment with constant rounds had been constructed from any one-way functions in last centuries, and the second with non-constant rounds were constructed from any one-way functions in FOCS2006, STOC2006 and STOC2007 respectively, furthermore, the lower bound of round complexity of statistically hiding commitments has been proven to be $\frac{n}{logn}$ rounds under the existence of one-way function.

Perfectly hiding commitments implies statistically hiding, hence, it is also infeasible to construct a practically perfectly hiding commitments with constant rounds under the existence of one-way function. In order to construct a perfectly hiding commitments with constant rounds, we have to relax the assumption that one-way functions exist. In this paper, we will construct a practically perfectly hiding commitment with two-round from any one-way permutation. To the best of our knowledge, these are the best results so far.

Keywords: Cryptography, perfectly hiding commitments, one-way permutation, Σ-protocol.

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[5, 8, 9, 10]. According to the computational power of senders and receivers, commitments can be classified into the four possible types shown in following Table[14].

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

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

[†] Email: tangcm622@hotmail.com

Usually, if the binding property holds with respect to a computationally unbounded algorithm sender, the commitment scheme is said to be *unconditionally binding*, else to be *computationally binding*; if instead, the hiding property holds with respect to a computationally unbounded algorithm receiver, the commitment scheme is said to be *unconditionally hiding*, else to be *computationally hiding.* Where "unconditionally binding (or hiding)" means "perfectly binding (or hiding)" or "statistically binding (or hiding)".

Feige and Shamir[6] constructed a commitment of Type A under the existence of one-way functions, and Goldreich et al. [9] also constructed the commitment scheme of Type C from any one-way functions. In [21], 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 [7], and then to any family of collision-resistant hash function [19]. In 1992, Naor et al. [17] showed that the collision resistance criterion is not necessary by giving a beautiful construction of perfectly hiding commitments with non-constant rounds from any one-way permutation, which is an 1-1 and length preserving one-way function. **it is left as an open question whether these primitives could be based on arbitrary one-way functions, which could again be essentially minimal** by [20, 22]. The progress in the past decade came in 2005 when Haitner et al. [12] 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. [16, 18] in 2006 and Haitner et al. [11] in 2007 constructed statistically hiding commitment from any one-way function respectively, however, their schemes need polynomial number of rounds complexity. Especially, Haitner et al.[11] gave the lower bound of round complexity of statistically hiding commitments which is $\frac{n}{logn}$ rounds under the existence of one-way function.

Perfectly hiding commitments implies statistically hiding, hence, it is also infeasible to construct a practically perfectly hiding commitments with constant rounds under the existence of one-way function. In order to construct a practically perfectly hiding commitments with constant rounds, we have to relax the assumption that one-way functions exist. In this paper, we will construct a practically perfectly hiding commitment with two-round from any one-way permutation, which is different from commitment with non-constant rounds from one-way permutation in [17]. 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[3]. So far, lots of cryptographic protocols have been constructed based on Σ-protocols, such as, identification schemes, digital signature schemes, secret sharing schemes[3].

Based on Σ-protocol, a new method to construct a commitment scheme was proposed in [13], 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 this paper, we use Σ-protocol as a main tool to construct *a perfectly hiding commitment with two-round from any one-way permutation.*

1.1 Our Contribution

We firstly construct a perfectly hiding and computationally binding commitment scheme with **two-round** under the existence of one-way permutation.

1.2 Related Works

Perfectly hiding commitments can be constructed from specific number-theoretic complexity assumptions [1, 2, 13]. In 1992, Naor et al. gave a beautiful construction of perfecly hiding commitments from any one-way permutation [17], however, their commitment has non-constant round complexisy. Haitner et al. constructed statistically hiding commitment from any one-way function [11], however, they also gave the lower bound of round complexity of statistically hiding commitments which is $\frac{n}{logn}$ rounds under the existence of any one-way function.

1.3 Organization

We start with some basic definitions and properties on Σ-protocol and commitment scheme. Then, we introduce a perfectly hiding commitment 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, $\nu : \mathbb{N} \rightarrow \mathbb{N}$ 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 $f : \{0,1\}^* \rightarrow \{0,1\}^*$ 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

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

Consider the case where f is an **one-way permutation** which is an 1-1 and length preserving one-way function

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[9, 14]. 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***[5] 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*

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

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

- 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

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

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

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

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

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

It is easy to show that perfect hiding commitment implies statistically hiding commitment, which in turn implies computationally hiding commitment.

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 *[13] The properties of Σ-protocols are invariant under parallel composition.*

3 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 [4] 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 perfect zero-knowledge argument (not proof) for honesty verifier. Hence, *perfectly hiding commitment scheme can be constructed from* Σ*-protocol with perfect zero-knowledge property.*

Now, we recall computational zero-knowledge argument (not proof) on Hamiltonian Cycle(in short HC) which is a NP-complete statement, which is revised from the protocol in [9] and [15].

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 C which exists if one-way function exist.*

Theorem 2 *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 [9].

Obviously, the protocol will be a Σ-protocol basing on 1), 2) and 3) if commitment scheme of Type C exist. It is well-known that commitment scheme of Type C exist if one-way function exist from [7, 9]. Hence, the above theorem follows.■

Protocol HC is also computational zero-knowledge argument of knowledge [9] 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 [6]:

Commitment Scheme 2

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 3 *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 $e = 1$, i.e., a is not independent of e. ■

Now, we obtain the fact that perfectly hiding or computationally hiding commitment scheme can be constructed from Σ-protocol. Hence, we can revise the result in [13] and obtain this following theorem.

Theorem 4 *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. ■

In fact, the above theorem implies if the commitment based on Σ-protocol will be a perfectly hiding commitment if the Σ-protocol is a perfect zero-knowledge argument (not proof).

4 Perfectly Hiding Commitment Scheme from Any One-Way Permutation

In order to obtain a perfectly hiding commitment scheme, we need introduce a protocol in 4.1.

4.1 A Computationally Binding Special Bit Commitment Scheme

We will propose a protocol between a computationally bounded sender P and a receiver V (with computationally bounded or unbounded power), which has computationally binding and has not hiding property. We emphasize our protocol is not a bit commitment scheme because it does not satisfy hiding, however, we cannot find a better name to denote it, so we call it as a special commitment scheme.

Computationally binding special bit commitment scheme

1. **Initialization**: P and V choose one-way permutation $f : \{0,1\}^k \to \{0,1\}^k$.
2. **Commit phase:**
   (a) To receive a commitment to a bit (using security parameter k, V uniformly selects $r \in \{0,1\}^k$ and sends it to P.
   (b) Upon receiving the message r (from V), P commits to value $b \in \{0,1\}$ by uniformly selecting $s \in \{0,1\}^k$ and sending $f(s)$ if $b = 0$, and $f(s \oplus r) \oplus r$ otherwise.
3. **Reveal phase:** In the reveal phase, P reveals the string s used in the commit phase. V accepts the value 0 if $f(s) = \alpha$ and accepts the value 1 if $f(s \oplus r) \oplus r = \alpha$, where (r, α) is the receiver's view of the commit phase.

Theorem 5 *The computationally binding special bit commitment scheme has the following properties: 1) computationally binding; 2) any computationally unbounded adversary can open the commitment α both as 0 and 1 with same probability.*

Proof: 1) (*computationally binding*) For any computationally bounded sender, if he can opened a commitment α both as 0 and 1, that is, he holds two s and s' such that $f(s) = f(s' \oplus r) \oplus r$ (or $f(s') = f(s \oplus r) \oplus r$), however, he holds them with negligible probability from f.

2) any computationally unbounded adversary can invert f, then he can compute s from $f(s)$ and s' from $f(s' \oplus r) \oplus r$, that is, he can open any α with length k as 0 or 1. In fact, he opens α as 0 or 1 with same probability because he can find sole s such that $f(s) = \alpha$ and sole s' such that $f(s' \oplus r) \oplus r = \alpha$ from one way permutation f. ■

4.2 A Perfectly Hiding Commitment

In order to obtain a perfectly hiding commitment, we will improve the commitment scheme 2 by the following steps:

1. In commitment scheme 2, message a with form $a_1 \circ a_2 \circ \cdots \circ a_{n^2}$ is commitment to every entry, which is 0 or 1, of the $n \times n$ adjacency matrix, where the bit commitment scheme is a perfectly binding and computationally hiding. We will use our special bit commitment scheme in 4.1 to replace the perfectly binding and computationally hiding commitment.

2. the prover should transfer the message a to another a', furthermore, any computationally unbounded receiver V can obtain every probable a' from a with same probability.

Now, we propose a commitment scheme 3 by improving commitment scheme 2 according to above steps, then prove it to be a perfectly hiding commitment scheme.

Bit 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 $e \in \{0, 1\}$

   (a) V selects uniformly $r_i \in \{0,1\}^k (1 \leq i \leq n^2)$ and sends them to P.

   (b) **Remark:** These random numbers are used to commit to entries of $n \times n$ adjacency matrix by using commitment scheme introduced in 4.1.

   (c) P runs the simulator M on input G, e to get (a, e, z).

      i. 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.

      ii. 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 cycle in this matrix.

      iii. message a has the form $a_1 \circ a_2 \circ \cdots \circ a_{n^2}$, where $a_i = f(s_i)$ *or* $f(s_i \oplus r_i) \oplus r_i$ and $|a_i| = k (1 \leq i \leq n^2)$.

   (d) **Remark:** (a, e, z) in this protocol is different from (a, e, z) in Commitment Scheme 2 because the commitment scheme used in message a is different.

   (e) P selects uniformly $R \in \{0,1\}^{n^2 k}$, computes $a' = a \oplus f_1(R)$, where $f_1 : \{0,1\}^{n^2 k} \rightarrow \{0,1\}^{n^2 k}$ is an one-way permutation.

   (f) P sends commitment a' to e to V.

3. **(Decommit)** To open the commitment a', P sends $(R, s_1, s_2, \cdots, s_{n^2}, \pi)$ to V if $e = 0$, and $(R, s'_1, \cdots, s'_n \in \{s_1, s_2, \cdots, s_{n^2}\}, \{a_1, a_2, \cdots, a_{n^2}\}/\{a'_1, \cdots, s'_n\})$ otherwise.

   (a) V accepts $e = 0$ if $a_i = f(s_i)$ *or* $a_i = f(s_i \oplus r_i) \oplus r_i (1 \leq i \leq n^2)$, $a' = a \oplus f_1(R)$ and the commitment a is commitment to the entries of the $n \times n$ adjacency matrix of $\pi(G)$ i.e., $(a, 0, z)$ is valid.

   (b) V accepts $e = 1$ if $a'_i = f(s'_i \oplus r_i) \oplus r_i (1 \leq i \leq n)$, $a' = a \oplus f_1(R)$, and n committed entries of commitments $a'_1, \cdots, a'_n$ is a cycle in this $n \times n$ matrix, i.e., $(a, 1, z)$ is valid.

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^* open commitment a' as followed:

(1): He opens the commitment a' both as $(R, s_1, s_2, \cdots, s_{n^2}, \pi)$, and as $(R, s'_1, \cdots, s'_n \in \{s_1, s_2, \cdots, s_{n^2}\}, \{a_1, a_2, \cdots, a_{n^2}\}/\{a'_1, \cdots, s'_n\})$.

(2): He opens the commitment a' both as $(R, s_{11}, s_{12}, \cdots, s_{1n^2}, \pi)$, and as $(R', s'_1, \cdots, s'_n \in \{s_1, s_2, \cdots, s_{n^2}\}, \{a_1, a_2, \cdots, a_{n^2}\}/\{a'_1, \cdots, s'_n\})$

In (1), it means that P^* can open commitment a' both as $e = 0$ by the validity of $(R, s_1, s_2, \cdots, s_{n^2}, \pi)$ as $e = 1$ by the validity of $(R, s'_1, \cdots, s'_n \in \{s_1, s_2, \cdots, s_{n^2}\}, \{a_1, a_2, \cdots, a_{n^2}\}/\{a'_1, \cdots, s'_n\})$, that is, he can obtain a Hamiltonian Cycle of grape G according to definition of Σ-protocol, which contradicts the assumption that HC is a NP-complete problem.

In (2), valid $(R, s_{11}, s_{12}, \cdots, s_{1n^2}, \pi)$ and $(R', s'_1, \cdots, s'_n \in \{s_1, s_2, \cdots, s_{n^2}\}, \{a_1, a_2, \cdots, a_{n^2}\}/\{a'_1, \cdots, s'_n\})$ are easily simulated, however it is impossible to obtain R, a_1, R', a such that $a_1 \oplus f_1(R) = a \oplus f_1(R')$ unless P^* can invert one-way function f, where $a_1 = a_{11} \circ a_{12} \circ \cdots \circ a_{1n^2}$ and $a = a_1 \circ a_2 \circ \cdots \circ a_{n^2}$ generated by $(s_{11}, s_{12}, \cdots, s_{1n^2})$ and $(s_1, s_2, \cdots, s_{n^2})$ respectively. However, P^* can invert function f with negligible probability.

For example, assume P can open commitment a' by using $(R, s_{11}, s_{12}, \cdots, s_{1n^2}, \pi)$, i.e. $e = 0$. Now, he look for $(R', s'_1, \cdots, s'_n \in \{s_1, s_2, \cdots, s_{n^2}\}, \{a_1, a_2, \cdots, a_{n^2}\}/\{a'_1, \cdots, s'_n\})$ in order to open a' as $e = 1$, that is, look for a pair (a, R') such that $a_1 \oplus f_1(R) = a \oplus f_1(R')$.

1. He may generate commitment a to a $n \times n$ matrix which is all 1, so he has to compute R' such that $a_1 \oplus f_1(R) = a \oplus f_1(R')$. Obviously, he can obtain R' with negligible probability.

2. He may select a random number $R' \in \{0, 1\}^{n^2 k}$, then computes $a = a_1 \oplus f_1(R) \oplus f_1(R')$. In order to propose a cycle with n nodes, he has to find $s'_1, \cdots, s'_n$ from a, that is, he has to find $s'_1, \cdots, s'_n$ to generate $a'_1, \cdots, a'_n$, where $a'_i = f(s'_i \oplus r'_i) \oplus r'_i$. However, P find $s'_1, \cdots, s'_n$ with negligible probability from one-way function f.

Similarly, it is impossible to find a pair (a_1, R) such that $a_1 \oplus f_1(R) = a \oplus f_1(R')$ if P holds a valid $(R', s'_1, \cdots, s'_n \in \{s_1, s_2, \cdots, s_{n^2}\}, \{a_1, a_2, \cdots, a_{n^2}\}/\{a'_1, \cdots, s'_n\})$.

Why should f be a one-way permutation? In fact, if one-way function f has collision intractable property, that is, the prover can find some pairs (r, r') such that $f(r) = f(r')$ and $r \neq r'$ or can find a r' from $f(r)$ such that $f(r) = f(r')$ and $r \neq r'$, the prover P^* maybe open the commitment a' both as 0 and 1 with non-negligible probability.

(Perfectly Hiding) (I) For any a with form $a_1 \circ a_2 \circ \cdots \circ a_{n^2}$, its every $a_i (1 \leq i \leq n^2)$, which is a commitment to an entry of a $n \times n$ matrix, can be opened as 0 or 1 with the same probability $\frac{1}{2}$ from theorem 5.

(I) For every a' with length $n^2 k$, the probability of any a with length $n^2 k$ to generate commitment a' is the same for any computationally unbounded adversary, because there always exists a R such that $a' = a \oplus f_1(R)$ for any pair (a, a').

From (I) and (II), every a' with length $n^2 k$ can be opened both as $e = 0$ and $e = 1$ with same probability if any computationally unbounded adversary knows a HC C of directed graph G. That is, the distribution of a' is independent of the value e. According to theorem 4, this commitment is perfectly hiding. ■

In *Bit Commitment Scheme 3*, the prover P must convince that graph G generated by the verifier V has Hamiltonian Cycle C by using a witness hiding protocol [6], 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 [23, 24], before P runs the simulator. Then, there are 3 moves in a Σ-protocol in the Setup phase, and there are 2 moves in the Commit phase. Hence, there are total 5 moves in in the Setup and Commit phase, however, we can regroup the third move in the Setup phase and the first move in the Commit phase into a move, as a result, a two-round commitment can be constructed. The commitment schemes can be composed in parallel from Proposition 1.

Theorem 7 *There exists a perfectly hiding and computationally binding commitment scheme with two rounds complexity.*

Theorem 8 *Perfectly hiding and computationally binding commitment scheme exists if one-way permutation exist.*

Proof(Sketch): *Commitment Scheme 3* is perfectly hiding and computationally binding from theorem 6. Commitment scheme used in the first message a in *Protocol HC* is commitment scheme of Type C which exists under existence of any one-way function. Both of f in computationally binding special commitment and f_1 in Commitment Scheme 3 are one-way permutations. Hence, perfectly hiding and computationally binding commitment scheme exists if one-way function exist. ■

Our commitment schemes have only one round complexity in the random oracle model[13].

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

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

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

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

[7] 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

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

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

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

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

[12] 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.

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

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

[15] 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

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

[17] 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.

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

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

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

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

[22] 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

[23] R. Pass. *Alternative Variants of Zero-Knowledge Proofs.* Licentiate Thesis, Stockholm, Sweden 2004

[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