An Efficient and Provably Secure Threshold Blind Signature

Jinho Kim, Kwangjo Kim, and Chulsoo Lee

IRIS (International Research center for Information Security)
ICU (Information and Communications University)
58-4 Hwaam-dong, Yusong-gu, Taejon, 305-732, S. Korea
Tel : +82-42-866-6118, FAX : +82-42-866-6154
{kman, kkj, csl100}@icu.ac.kr

Abstract. We propose an efficient and provably secure threshold blind signature scheme based on Okamoto-Schnorr blind signature. In our scheme, any t $(t < n)$ out of n signers can generate a blind signature and any one can verify the signature using the same verification process used in Okamoto-Schnorr blind signature. New scheme can be used for blind signature-based voting systems with multiple administrators and secure electronic cash systems to prevent their abuse. We prove that our scheme is to be as secure as Okamoto-Schnorr's blind signature scheme and more efficient than Juang *et al.*'s scheme [12].

1 Introduction

The concept of a blind signature scheme was initially proposed by Chaum [2] in 1982. A blind signature scheme is two party protocol between a sender A and a signer B. The basic idea is as follows: A sends a piece of information m to B in a blinded form. Then B signs and returns it to A. From this signature, A can unblind and compute B's signature on message m. At the completion of the protocol, B knows neither the message m nor the signature associated with it. The purpose of a blind signature is to prevent the signer B from observing the message he signs and its resulting signature. It is widely used in many applications like secure voting protocol and electronic payment systems [1, 3] when anonymity of a message is required.

In general, the voting system based on blind signature scheme [7, 6, 14] is managed by a single administrator, who can be empowered to authorize votes. But, a dishonest administrator can abuse this power to cast fraudulent votes. To prevent this abuse by a single administrator, we need more than one administrator using threshold signature scheme to sign a vote. By using our scheme, the power of a single administrator can be distributed to n administrators and any t $(t < n)$ out of n administrators can generate a signature for a given message.

Juang and Lei initially proposed (t, n) threshold blind signature scheme in [11], and a provably secure threshold blind signature scheme [12] based on Okamoto-Schnorr blind signature. But, their work takes more time in the partial signature generation and verification processes. A user A and each signer

P_i in [12] perform $O(n)$ and $O(n)$ multiplications, whereas A and P_i in our scheme perform $O(t)$ and $O(1)$ multiplications. In our scheme, we use the partial public key published by each signer P_i in key generation phase to reduce the computational complexity in the signature generation phase.

In this paper, we propose an efficient and provably secure threshold blind signature scheme and prove its security in the random oracle model. We will show the proposed scheme is as secure as Okamoto-Schnorr's blind signature scheme and is more efficient than [12]. The signature size and verification process in our scheme are the same as those of Okamoto-Schnorr's blind signature scheme.

This paper is organized as follows: In Section 2, we will review the related works such as blind signature, threshold signature, and distributed key generation. Then we propose our scheme in Section 3. In Section 4, we define some notions of security and prove the security of our scheme. The performance analysis is presented in Section 5. Finally, concluding remarks are made in Section 6.

2 Related Works

2.1 Blind Signature Scheme

Parameters
Suppose that p and q are two large primes such that $q|p-1$. g and h are elements of Z_p^* with order q. We assume that it is infeasible to compute the integer d such that $g = h^d \bmod p$, given g, h, and p.

Okamoto-Schnorr Blind Signature

This blind signature scheme designed by Okamoto [13] is an extension of the Schnorr scheme in [19]. An entity A creates a secret key $(r, s) \in Z_q \times Z_q$ and publishes the public key, $y = g^r h^s \bmod p$. Two party protocols between a sender A and a signer B are as follows:

1. B chooses $t, u \in Z_q$, and sends $a = g^t h^u \bmod p$ to A.
2. A blinds a with $\beta, \gamma, \delta \in Z_q$ into $\alpha = a g^\beta h^\gamma y^\delta \bmod p$, computes $\varepsilon = H(m, \alpha)$ and sends $e = \varepsilon - \delta \bmod q$ to B.
3. B computes $R = t - er \bmod q$ and $S = u - es \bmod q$, and sends (R, S) which satisfies $a = g^R h^S y^e \bmod p$.
4. A computes $\rho = R + \beta \bmod q$ and $\sigma = S + \gamma \bmod q$.
5. Issue a signature (α, ρ, σ) for message m.
6. We can easily verify that $\alpha = g^\rho h^\sigma y^\varepsilon \bmod p$, where $\varepsilon = H(m, \alpha)$.

Okamoto-Schnorr blind signature was proved to be provably secure under adaptively chosen message attacks in the random oracle model [17, 18].

2.2 Threshold Signature Scheme

In 1979, Shamir at first defined the notion of threshold scheme [20] (often called a secret sharing scheme). The (t, n) threshold scheme proposed by Shamir is to divide a secret D into n pieces $D_1, D_2, \ldots, D_n$ in such a way that:

- Knowledge of any t $(t < n)$ or more D_i pieces makes D easily computable.
- Knowledge of any $t-1$ or fewer D_i pieces leaves D completely undetermined.

Threshold signature schemes have received considerable attention in the literature [4, 8, 10, 16, 21]. In a threshold signature scheme, digital signatures can be issued by t players rather than by one party. The secret key should be shared by n players in advance, and any set of less than t players cannot issue a digital signature for a given message.

2.3 Distributed Key Generation

Distributed key generation (DKG) defined in [9] is a secret sharing protocol without any trusted party (dealer). DKG works as a main component of threshold cryptosystems. Feldman [5] proposed a verifiable secret sharing (VSS) protocol based on discrete logarithm problem. But, his protocol has been shown to have a security flaw, since an adversary can influence the distribution of the result of his protocol to a non-uniform distribution. Moreover, Pedersen [15] proposed the first DKG protocol based on Feldman's protocol, and Gennaro *et al.* [9] proposed a provably secure DKG protocol. We will explain the DKG protocol in brief.

Each player P_i performs the following steps:

1. Each player chooses two random polynomials f_i, f'_i over Z_p where $f_i(x) = a_{i0} + a_{i1}x + \cdots + a_{it-1}x^{t-1}$, $f'_i(x) = b_{i0} + b_{i1}x + \cdots + b_{it-1}x^{t-1}$. Let $r_i = a_{i0} = f_i(0)$ and $r'_i = b_{i0} = f'_i(0)$. P_i broadcasts $C_{ik} = g^{a_{ik}} h^{b_{ik}} \bmod p$, for $k = 0, \ldots, t-1$. P_i computes the shares $s_{ij} = f_i(j) \bmod p, s'_{ij} = f'_i(j) \bmod p$, for $j = 1, \ldots, n$ and sends (s_{ij}, s'_{ij}) secretly to player P_j.
2. Each player P_j verifies the shares (s_{ij}, s'_{ij}). P_j checks if

$$g^{s_{ij}} h^{s'_{ij}} = \prod_{k=0}^{t-1} (C_{ik})^{j^k} \bmod p. \tag{1}$$

   If the check fails for an index i, P_j broadcasts a complaint against P_i.
   (a) Each player P_i who received a complaint defends himself by broadcasting the value (s_{ij}, s'_{ij}) that satisfies Eq. (1).
   (b) Any player who either received more than $t-1$ complaints, or answered wrong value to a complaint is disqualified.
   (c) Let $H_0 := \{P_j | P_j \text{ is a not disqualified player}\}$.
   (d) The distributed secret value $r = \sum_{j \in H_0} r_i$ is not computed by any player.
3. Extracting $y = g^r \bmod p$:
   (a) Each player $P_j \in H_0$ broadcasts $A_{jk} = g^{a_{jk}} \bmod p$, for $k = 0, \ldots, t-1$

(b) Each player P_j verifies the values broadcasted by other players in H_0, namely, for each $i \in H_0$, P_j checks if

$$g^{s_{ij}} = \prod_{k=0}^{t-1} (A_{ik})^{j^k} \bmod p. \tag{2}$$

If the check fails for an index i, P_j complains against P_i by broadcasting the value (s_{ij}, s'_{ij}) that satisfies Eq. (1) but does not satisfy Eq. (2).

(c) For players P_i who received at least one valid complaint, i.e., value which satisfies Eq. (1) but does not satisfy Eq. (2), the other players run the reconstruction phase of Pedersen's VSS to compute $r_i, f_i(\cdot), A_{ik}$ for $k = 0, \ldots, t-1$

3 Threshold Blind Signature Scheme

3.1 Key Generation Protocol

We will show how to generate a random shared secret. By using the protocol presented in section 2.3, a random shared secret can be generated. But, since our scheme employs Okamoto-Schnorr blind signature scheme, we need two different secret keys. So we use a modified version of DKG protocol.

Each player P_i performs the following steps to generate a random shared secret:

1. Each player chooses three random polynomials f_i, f'_i, f''_i over Z_p where $f_i(x) = a_{i0} + a_{i1}x + \cdots + a_{it-1}x^{t-1}$, $f'_i(x) = a'_{i0} + a'_{i1}x + \cdots + a'_{it-1}x^{t-1}$, and $f''_i(x) = b_{i0} + b_{i1}x + \cdots + b_{it-1}x^{t-1}$. Let $a_{i0} = f_i(0), a'_{i0} = f'_i(0)$, and $b_{i0} = f''_i(0)$. P_i broadcasts $C_{ik} = g^{a_{ik}} h^{b_{ik}} \bmod p$ and $D_{ik} = g^{b_{ik}} h^{a'_{ik}} \bmod p$ for $k = 0, \ldots, t-1$. P_i computes the shares $s_{ij} = f_i(j) \bmod p, s'_{ij} = f'_i(j) \bmod p$, and $s''_{ij} = f''_i(j) \bmod p$ for $j = 1, \ldots, n$ and sends $(s_{ij}, s'_{ij}, s''_{ij})$ secretly to player P_j.
2. Each player P_j verifies the shares $(s_{ij}, s'_{ij}, s''_{ij})$. P_j checks if

$$g^{s_{ij}} h^{s''_{ij}} = \prod_{k=0}^{t-1} (C_{ik})^{j^k} \bmod p \ , \ g^{s''_{ij}} h^{s'_{ij}} = \prod_{k=0}^{t-1} (D_{ik})^{j^k} \bmod p. \tag{3}$$

   If the check fails for an index i, P_j broadcasts a complaint against P_i. Then, P_i and P_j perform the same works as they do step2 (a)-(c), in section 2.3.
3. Extracting $y = g^s h^{s'} \bmod p$:
   (a) Each player $P_i \in H_0$ broadcasts $A_{ik} = g^{a_{ik}} h^{a'_{ik}} \bmod p$ for $k = 0, \ldots, t-1$.
   (b) Each player P_j checks if

$$g^{s_{ij}} h^{s'_{ij}} = \prod_{k=0}^{t-1} (A_{ik})^{j^k} \bmod p. \tag{4}$$

   if the checks fails for an index i, P_j complains against P_i by broadcasting the value $(s_{ij}, s'_{ij}, s''_{ij})$ that satisfies Eq. (3), but does not satisfy Eq. (4).

(c) Each player P_i who received a complaint defends himself by broadcasting the value $(s_{ij}, s'_{ij}, s''_{ij})$ that satisfies Eq. (4).
(d) Any player who either received more than $t-1$ complaints, or answered wrong value to a complaint is disqualified.

After executing this protocol, the following equations hold :

$$y = \prod_{j \in H_0} g^{a_{j0}} h^{a'_{j0}} = \prod_{j \in H_0} A_{j0}$$
$$f(x) = a_0 + a_1 x + \cdots + a_{t-1} x^{t-1}, \text{ where } a_i = \sum_{j \in H_0} a_{ji}$$
$$f'(x) = a'_0 + a'_1 x + \cdots + a'_{t-1} x^{t-1}, \text{ where } a'_i = \sum_{j \in H_0} a'_{ji}$$
$$f(i) = s_i, f'(i) = s'_i, \forall i = 1, \ldots, n$$

For convenience, let $|H_0| = n$ and the key generation protocol be denoted by

$$\{(s_1, s'_1), \ldots, (s_n, s'_n)\} \longleftrightarrow \{(a_0, a'_0) \mid y, y_j, H_0\}, \text{ for } j = 1, \ldots, n$$

, where H_0 means an index set of disqualified players in key generation protocol, and (a_0, a'_0) is a secret key and y is a public key. For each $j \in H_0$, the value $y_j = g^{s_j} h^{s'_j}$ is the public commitment and (s_j, s'_j) is the secret key share of P_j.

3.2 Signature Generation Protocol

Let m be a message and let H be a one-way hash function. Suppose that a user A wants to get a signature from t signers, $H_1 \subseteq H_0$. For simplicity, let $H_1 = \{1, \ldots, t\}$ and let the output of key generation protocol be as follows :

$$\{(r_1, s_1), \ldots, (r_n, s_n)\} \longleftrightarrow \{(r, s) \mid y, y_j, H_0\}, \text{ for } j = 0, \ldots, n$$

The signature generation protocol starts when a user A requests each signer P_i, where $i \in H_1$, threshold blind signature by sending $\omega_i = \prod_{k=1, k \neq i}^{t} \frac{k}{k-i}$ for $k \in H_1$. A and P_i perform the following protocol:

1. Each P_i chooses $t_i, u_i \in Z_q$, and sends $a_i = g^{t_i} h^{u_i} \bmod p$ to A.
2. After receiving all a_i's, A computes $a = \prod_{i=1}^{t} a_i$ and blinds a with $\beta, \gamma, \delta \in Z_q$ into $\alpha = a g^{\beta} h^{\gamma} y^{\delta} \bmod p$. A computes $\varepsilon = H(m, \alpha)$ and sends $e = \varepsilon - \delta \bmod q$ to each P_i.
3. After receiving e, each P_i computes $R_i = t_i - e r_i \omega_i \bmod q$ and $S_i = u_i - e s_i \omega_i \bmod q$, and sends (R_i, S_i) which satisfies $a_i = g^{R_i} h^{S_i} y_i^{e\omega_i} \bmod p$.
4. A checks $a_i = g^{R_i} h^{S_i} y_i^{e\omega_i}$ after receiving (R_i, S_i). If the check fails for an index i, A sends e again to P_i to take a correct (R_i, S_i). Otherwise, A makes a signature (α, ρ, σ), where $\rho = \beta + \sum_{i=1}^{t} R_i \bmod q$, $\sigma = \gamma + \sum_{i=1}^{t} S_i \bmod q$.

3.3 Signature Verification

The verification process is the same as Okamoto-Schnorr blind signature scheme. A verifier accepts a signature (α, ρ, σ) on a message m if and only if

$$\alpha = g^{\rho} h^{\sigma} y^{\varepsilon} \bmod p, \text{ where } \varepsilon = H(m, \alpha).$$

3.4 Correctness

The validity of the signature on message m can be checked easily as follows:

$$\begin{aligned}
\alpha &\equiv g^{\rho} h^{\sigma} y^{\varepsilon} \\
&\equiv g^{\beta + \sum_{i=1}^{t} R_i} h^{\gamma + \sum_{i=1}^{t} S_i} y^{\varepsilon} \\
&\equiv g^{\beta} h^{\gamma} (g^{\sum_{i=1}^{t} t_i - er} h^{\sum_{i=1}^{t} u_i - es}) y^{\varepsilon} \\
&\equiv g^{\beta} h^{\gamma} (g^{\sum_{i=1}^{t} t_i} h^{\sum_{i=1}^{t} u_i})(g^{-er} h^{-es} y^{\varepsilon}) \\
&\equiv a g^{\beta} h^{\gamma} y^{\varepsilon - e} \\
&\equiv a g^{\beta} h^{\gamma} y^{\delta} \bmod p.
\end{aligned}$$

4 Security

In this section, we will show that our scheme is secure if and only if Okamoto-Schnorr blind signature scheme is secure, i.e., the security of the two scheme is identical.

4.1 Notions of Security

The notions of security for blind signature scheme were formally defined in [17, 12] under the random oracle model.

One-More Forgery : For some polynomially bounded integer ℓ, an attacker can obtain $\ell + 1$ valid signatures after fewer than ℓ interactions with the signer.

Parallel Attack : the attacker interacts ℓ times in parallel with the signer.

We define adversaries and the notions of security for Okamoto-Schnorr blind signature scheme and (t, n) threshold blind signature scheme. Let $\tilde{A}_{OS}$ be an adversary who is allowed to execute Okamoto-Schnorr blind signature protocol. $\tilde{A}_{OS}$ can ask a signer to generate blind signatures on chosen messages. Let $\tilde{A}_D$ be an adversary who is allowed to execute the signature generation protocol. $\tilde{A}_D$ can ask any t signers to generate blind signatures on the chosen messages. $\tilde{A}_D$ also might corrupt up to $t-1$ arbitrary signers.

Definition 1. *A blind signature scheme is secure, if no* $\tilde{A}_{OS}$ *can do the one-more forgery with non-negligible probability in the random oracle model under parallel attack.*

Definition 2. *A* (t, n) *threshold blind signature scheme is secure, if no* $\tilde{A}_D$ *can do the one-more forgery with non-negligible probability in the random oracle model under parallel attack.*

4.2 Proof of Security

First, we will analyze the adversary's view during the generation of a random shared secret, and build a simulator SIM, such that produces an output distribution which is polynomially indistinguishable from $\tilde{A}_D$'s view for a given public key y. Let B be the index set of corrupted players and B' be the index set of the players who publishes inconsistent values A_{im}.

Theorem 1. *For every probabilistic polynomial time adversary* $\tilde{A}_D$*, there exists a probabilistic polynomial time simulator* SIM*, such that for given* y*, produces an output distribution which is indistinguishable from* $\tilde{A}_D$*'s view.*

Proof. (sketch) In the key generation protocol, $\tilde{A}_D$ sees the following probability distribution of data produced by the uncorrupted parties :

1. The content of the random tape of $\tilde{A}_D$
2. Sharing polynomials : $f_i(\cdot), f_i'(\cdot), f_i''(\cdot)$ for $i \in B$
3. Shares : $s_j(i), s_j'(i), s_j''(i)$ for $j \in H_0, i \in B$
4. Public commitments in step 1: C_{jk}, D_{jk} for $j \in H_0, k = 1, \ldots, t-1$
5. Public commitments in step 3: A_{jk} for $j \in H_0, k = 1, \ldots, t-1$
6. Answers on a valid complaint : $s_{ij}, s_{ij}', s_{ij}''$ for $i \in B', j = 1, \ldots, n$

We use the same simulator SIM that can act as the real players in the protocol as in [9, 22]. A simulator SIM can be constructed as follows:

1. Since (1) and (2) are internal values of adversary, SIM does not have to compute these values.
2. SIM performs steps 1 and 2 in subsection 3.2 on behalf of the uncorrupted players. After this step, SIM knows all shares $s_j(i), s_j'(i)$, and $s_j''(i)$ in (3) and public values in (4).
3. To compute A_{ik} for $i \in H_0, k = 1, \ldots, t-1$, SIM performs the following computations:
   - Compute $A_{ik} = g^{a_{ik}}$ for $i \in H_0 \backslash \{n\}, k = 0, \ldots, t-1$
   - Compute $A_{n0} = y \cdot \prod_{i \in H_0 \backslash \{n\}} (A_{i0})^{-1} \bmod p$
   - Compute $A_{nk} = (A_{n0})^{\lambda_{k0}} \cdot \prod_{i=1}^{t-1} (g^{s_{ni}})^{\lambda_{ki}}$ for $k = 1, \ldots, t-1$, where λ_{ki}'s are the Lagrange interpolation coefficients.
   - Broadcast A_{ik} for $i \in H_0, k = 1, \ldots, t-1$
   - $y = g^{a_{i0}} h^{a'_{i0}} = \prod_{j \in H_0} A_{j0}$

4. To handle the messages resulting from complaints, SIM acts as follows:
   - Perform for each uncorrupted player the verifications of Eq. (4) on the values A_{ik} for $i \in B$, broadcast by the players controlled by the adversary. If the verification fails for some $i \in B$ and $j \in H_0 \backslash B$, broadcast a complaint $(s_{ij}, s'_{ij}, s''_{ij})$.
   - For each valid complaint against P_i, perform the reconstruction phase of Pedersen's VSS to compute $f_i(0), f'_i(0)$ and y_i. □

A more detailed analysis of the distribution can be found in [9, 22].

Theorem 2. *For every adversary $\tilde{A}_{OS}$ who can produce one-more forgery under parallel attack, there exists an adversary $\tilde{A}_D$ who can produce one-more forgery under parallel attack.*

Proof. We can make $\tilde{A}_D$ given the adversary $\tilde{A}_{OS}$. Suppose the key generation protocol of $\tilde{A}_D$ generates a public key y. Whenever $\tilde{A}_{OS}$ asks for a signature on a message m, $\tilde{A}_D$ executes some t signers to issue a signature and returns the signature to $\tilde{A}_{OS}$. Since $\tilde{A}_{OS}$ can perform its parallel attack, $\tilde{A}_{OS}$ can produce one-more forgery. Finally, $\tilde{A}_D$ takes the output of $\tilde{A}_{OS}$ and produces one-more forgery. Hence, if $\tilde{A}_{OS}$ outputs one-more forgery under parallel attack, $\tilde{A}_D$ can output one-more forgery. □

It is known that Okamoto-Schnorr blind signature scheme is provably secure in the random oracle model [17]. Hence, we can say that there is no $\tilde{A}_{OS}$ that can produce one-more forgery under parallel attack.

Theorem 3. *If any adversary $\tilde{A}_D$ can produce one-more forgery under parallel attack, there exists an adversary $\tilde{A}_{OS}$ who can produce one-more forgery under parallel attack.*

Proof. We show how to construct $\tilde{A}_{OS}$ given the adversary $\tilde{A}_D$. For simplicity, assume $\tilde{A}_D$ corrupts players $1, \ldots, t-1 \in H_1$. $\tilde{A}_{OS}$ executes SIM described in **Theorem 1** to perform key generation protocol for a given public key y, and then runs $\tilde{A}_D$. Whenever $\tilde{A}_D$ needs a signature of m_i, $\tilde{A}_{OS}$ interacts with signer to obtain a signature of m_i. $\tilde{A}_{OS}$ has to provide $\tilde{A}_D$ the signature and the view that he sees during the signature generation protocol. $\tilde{A}_{OS}$ performs steps 1 and 2 on behalf of the players $P_1, \ldots, P_{t-1}$. After this simulation, $\tilde{A}_{OS}$ knows (a_j, R_j, S_j) for $j = 1, \ldots, t-1$. And then, $\tilde{A}_{OS}$ computes R_t, S_t, and a_t as follows.

$$\rho_i = \beta + \sum_{j=1}^{t} R_j, \ \ \sigma_i = \gamma + \sum_{j=1}^{t} S_j.$$

R_t, S_t, and a_t are computed as

$$R_t = \rho_i - \beta - \sum_{i=1}^{t-1} R_i,\ S_t = \sigma_i - \gamma - \sum_{i=1}^{t-1} S_i$$
$$a_t = g^{R_t} h^{S_t} y_t^{e\omega_t} \bmod p$$

$\tilde{A}_{OS}$ inputs (a_t, R_t, S_t) into $\tilde{A}_D$. Since the whole view of $\tilde{A}_D$ is given, $\tilde{A}_D$ can produce one-more forgery. Finally, $\tilde{A}_{OS}$ takes the output of $\tilde{A}_D$, and produces his one-more forgery. □

Corollary. *The proposed threshold blind signature scheme is secure if and only if Okamoto-Schnorr blind signature scheme is secure.*

Proof. The proof of this corollary follows immediately from **Theorems 1**, **2** and **3**. □

5 Performance

In this section, we compare the computational load required to perform threshold blind signature.

Table 1. Computational cost of the signature generation

	User A			Singer P_i		
	ADD	MUL	EXP	ADD	MUL	EXP
OS Scheme	3	3	3	2	3	2
JL Scheme	$2t+1$	$2n-t+6$	8	$2(n-t+1)$	$2n-1$	2
Our Scheme	$2t+1$	$t+5$	6	2	5	2

OS scheme : Okamoto-Schnorr blind signature scheme
JL scheme : Blind threshold scheme proposed by Lei and Juang
ADD: the number of additions
MUL: the number of multiplications
EXP : the number of exponentiations

Table 1 illustrates the cost of computations of blind signature scheme and threshold blind schemes. To reduce computational cost, we suppose that the value ω_i (for $i = 1, \ldots t$) is computed before the signature generation phase. In this case, the number of additions and multiplications done by each signer P_i in our scheme are the same as those of OS scheme. JL scheme takes more time in the partial signature generation and verification processes. In the JL scheme, the partial signature generation and verification are as follows :

– Partial signature generation
$R_i = e(r_i + \omega_i + \sum_{j=t+1}^{n} f_j(i)) + t_i \bmod p$
$S_i = e(s_i + \omega_i + \sum_{j=t+1}^{n} f'_j(i)) + u_i \bmod p$

– Partial signature verification
$g^{R_i} h^{S_i} y^e \equiv a_i(\prod_{j=t+1}^{n} g^{a_{ik}})^{\omega_i e}(\prod_{j=t+1}^{n} h^{a'_{ik}})^{\omega_i e} \bmod p$

A user A and each signer P_i in our scheme perform $O(t)$ and $O(1)$ multiplications, whereas A and P_i in JL scheme perform $O(n)$ and $O(n)$ multiplications. Compared with the JL scheme, our scheme is more efficient and practical. The signature verification cost of JL scheme and our scheme are the same as that of OS scheme.

6 Conclusion

We have proposed a secure and practical threshold blind signature scheme based on Okamoto-Schnorr blind signature scheme. In our scheme, any t out of n signers can generate a blind signature and any one can verify the signature using the same verification process of Okamoto-Schnorr scheme. We have proved the proposed scheme is as secure as Okamoto-Schnorr's blind signature scheme in the random oracle model. Moreover, we claim that our scheme is more efficient and practical than [12].

References

1. D. Chaum, *Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms*, Comm. ACM 24, pp.84–88, 1981.
2. D. Chaum, *Blind Signatures for Untraceable Payments*, Advances in Cryptology-Crypto'82, pp.199–203, Springer-Verlag, 1983.
3. D. Chaum, A. Fiat, M. Naor, *Untraceable Electronic Cash*, Advances in Cryptology-Crypto'88, LNCS 403, pp.319–327, Springer-Verlag, 1988.
4. Y. Desmedt, *Threshold Cryptosystems*, Advances in Cryptology-Auscrypt'92, LNCS 718, Springer-Verlag, 1993.
5. P. Feldman, *A Practical Scheme for Non-interactive Verifiable Secret Sharing*, In Proc. 28th FOCS, pp. 427-437, 1987.
6. A. Fujioka, T. Okamoto and K. Ohta, *A Practical Secret Voting Scheme for Large Scale Election*, Advances in Cryptology-Auscrypt'92, LNCS Vol.718, pp.248–259, Springer-Verlag, 1993.
7. A. Fujioka, M. Abe, M. Ohkubo, and F. Hoshino, *An Implementation and an Experiment of a Practical and Secure Voting Scheme*, Proc. of SCIS2000, C48, Okinawa, Japan, Jan. 26–28, 2000.
8. R. Gennaro, S. Jarecki,H. Krawczyk, and T. Rabin, *Robust Threshold DSS Signatures*, Advances in Cryptology-Eurocrypt'96, LNCS Vol.1070, pp.354–371, Springer-Verlag, 1996.
9. R. Gennaro, S. Jarecki,H. Krawczyk, and T. Rabin, *Secure Distributed Key Generation for Discrete-Log Based Cryptosystems*, Advances in Cryptology-Eurocrypt'99, LNCS Vol.1592, pp.295–310, Springer-Verlag, 1999.

10. S. Jarecki and A. Lysyanskaya, *Adaptively Secure Threshold Cryptography*, Advances in Cryptology-Eurocrypt'00, LNCS Vol.1807, pp.221–242, Springer-Verlag, 2000.
11. W. S. Juang and C. L. Lei, *Blind Threshold Signatures Based on Discrete Logarithm*, Proceedings of the 2nd Asian Computing Science Conference, Lecture Notes in Computer Science 1179, Springer-Verlag, pp. 172-181, 1996.
12. W. S. Juang, C. L. Lei, and Pei-Ling Yu, *Provably Secure Blind Threshold Signatures Based on Discrete Logarithm*, Proceedings of 1999 National Computer Symposium, pp. C198-C205, 1999.
13. T. Okamoto, *Provably Secure and Practical Identification Schemes and Corresponding Signature Schemes*, Advances in Cryptology-Crypto'92, LNCS Vol.740, pp.31–53, Springer-Verlag, 1992.
14. M. Ohkubo, F. Miura, M. Abe, A. Fujioka and T. Okamoto, *An Improvement on a Practical Secret Voting Scheme*, Information Security'99, LNCS Vol.1729, pp.225–234, Springer-Verlag, 1999.
15. T.P. Pedersen, *Non-interactive and Information-theoretic Secure Verifiable Secret Sharing*, Advances in Cryptology-Crypto'91, LNCS Vol.576, pp.129–140, Springer-Verlag, 1992.
16. T.P. Pedersen, *A Threshold Cryptosystem without a Trusted Party*, Advances in Cryptology-Eurorypt'91, LNCS Vol.547, pp.522–526, Springer-Verlag, 1992.
17. D. Pointcheval and J. Stern, *Provably Secure Blind Signature Scheme*, Advances in Cryptology-Asiacrypt'96, LNCS Vol.1163, pp.252–265, Springer-Verlag, 1996.
18. D. Pointcheval and J. Stern, *Security Arguments for Digital Signatures and Blind Signatures*, Journal of Cryptology, LNCS Vol.13, Num.3, pp.361–396, Springer-Verlag, 2000.
19. C. P. Schnorr, *Efficient Identification and Signatures for Smart Cards*, Advances in Cryptology-Crypto'89, LNCS Vol.435, pp.235–251, Springer-Verlag, 1990.
20. A. Shamir, *How to Share a Secret*, Comm. ACM Vol.22, pp.612–613, 1979.
21. V. Shoup, *Practical Threshold Signatures*, Advances in Cryptology-Eurocrypt'00, LNCS Vol.1807, pp.207–220, Springer-Verlag, 1996.
22. D.R. Stinson and R. Strobl, *Provably Secure Distributed Schnorr Signatures and a (t, n) Threshold Scheme for Implicit Certificates*, Information Security and Privacy, ACISP'01, LNCS Vol.2119 , pp. 417-434, Springer-Verlag, 2001.