Provably Secure Blind Threshold Signatures Based on Discrete Logarithm

Chin-Laung Lei, Wen-Shenq Juang and Pei-Ling Yu

Department of Electrical Engineering, Rm. 343
National Taiwan University
Taipei, Taiwan, R.O.C.
[*lei*, *vimal*, *bey*]@fractal.ee.ntu.edu.tw .

Abstract

In this paper, we propose a provably secure group-oriented blind (t,n) threshold signature scheme which is the first scheme, such that, its security is proved as equivalent as the discrete logarithm problem in the random oracle model. By the scheme, any t out of n signers in a group can represent the group to sign blind threshold signatures, which can be used in anonymous digital e-cash systems or secure voting systems. By our proposed scheme, the issue of e-coins is controlled by several authorities. In our scheme, the size of a blind threshold signature is the same as that of an individual blind signature and the signature verification process is equivalent to that of an individual signature.

Keywords: Provably Secure Blind Signatures, Threshold Signatures, Discrete Logarithm, Secure E-cash Systems, Secure Voting Systems.

1 Introduction

The concept of blind signature was introduced by Chaum [4]. A blind signature scheme is an interactive protocol which involves two kinds of participants, the signer and a requester. A distinguishing property required by a typical blind signature scheme [2, 4, 12, 25, 26] is so-called the "unlinkability", which ensures that requesters can prevent the signer from deriving the exact correspondence between the actual signing process performed by the signer and the signature which later made public. The blind signatures can realize the secure electronic payment schemes [4, 5, 7] protecting customers' anonymity, and the secure voting schemes [14, 15, 29] preserving voters' privacy. In a distributed environment, every signed blind message can be thought as a fixed amount of electronic money in secure electronic payment schemes, or as a ticket in applications such as secret voting schemes. In [25], two provably secure blind signature schemes are proposed. One has been proved to be equivalent to the discrete logarithm problem in a subgroup. The other has been proved to be equivalent to the RSA problem. In [26], a blind signature scheme is proposed and proved to be equivalent to factorization.

Threshold signatures [8, 11] are motivated by the need that arises in organizations to have a group of employees who agree on a message before signing and by the need to protect the group private key from the attack of internal and external adversaries. The later becomes more important with the actual deployment of public key schemes in practice. The signing power of some authorities inevitably invites attackers to try and steal this power. The goal of a threshold signature scheme is to increase the availability of the signing authority and to increase the protection against forgery by making it harder for the adversary to learn the group secret key.

To the date, the on-line e-cash schemes proposed in [4, 5] are more efficient and practical. The aim of these schemes was to produce an electronic version of money which retains the same properties as paper cash. In real world environments, if the issue of e-coins is controlled by a single person. He can generate extra e-coins as he wishes. To cope with this dilemma, instead of a unique administrator, every customer needs to request blind threshold signatures [16, 17] as e-coins from t arbitrary administrators, so that, t arbitrary administrators can represent the bank to issue e-coins.

All meta-blind threshold signature schemes [16, 17] have not been proven to be secure as some hard problems, e.g., the discrete logarithm problem. In this paper, we propose a provably secure blind threshold signature scheme which is the first scheme, such that, its security is proved to be equivalent to the discrete logarithm problem. Our proposed scheme can be directly applied to secure e-cash schemes for distributing the power of a single authority. The modified e-cash schemes can meet the real world environments without a single trusted authority or with some absent/dishonest authorities. In our scheme, the size of a blind threshold signature is the same as that of an individual blind signature and the verification process of a blind threshold signature is equivalent to that of an individual blind signature. Thus, our proposed scheme is optimal with respect to the blind threshold signature size and the verification process.

The paper is organized as follows. In Section 2, we present the definition of blindness of a threshold signature scheme and that of unforgeability of blind threshold signatures. In Section 3, we present a provably secure blind threshold signature

scheme. Then we discuss its correctness, security and performance in Section 4. Finally, a concluding remark is given in Section 5.

2 Preliminary

In this section, we present the definition of blindness of a threshold signature scheme and that of unforgeability of blind threshold signatures. There are two methods for verifying the validity of a signature: the comparison method and the restoration (message recovery) method [23]. In the comparison method, for verifying a signature, the corresponding message must be sent to a verifier along with the signature. To save the length of the signature, instead of signing the whole message, one can make a signature on the digest of the message which is the hashed value of a secure one-way hash function [20] with the message as input. In the restoration method, only the signature is sent to a verifier. The signed message which is embedded in the signature can be recovered after the verification process. Many signature schemes with message recovery have been proposed [22, 27].

Given a secret δ, we say that the secret shadows $(\delta_i, 1 \leq i \leq n)$ construct a (t, n) threshold secret sharing of δ if $t-1$ (or less) of these values reveal no information about δ and there exists a poly-time algorithm that outputs δ having any subset of t values as inputs.

Let there be $n > 1$ players in a distributed system and player i has his own secret s_i. A secure computing protocol for this system is a procedure for evaluating the function value $f(s_1, s_2, ..., s_n)$ jointly by the n players such that the output becomes commonly known while s_i remains secret. A secure computing protocol can be used to define blind threshold signature schemes. We define the blindness of a (t, n) threshold signature scheme with the comparison method as follows:

Definition 1 A blind (t, n) threshold signature scheme with the comparison method is a 12-tuple $\mathcal{P}_T = (\mathcal{M}, \mathcal{S}, \Delta, \mathcal{K}, \Lambda, \Psi, \Re, \Omega_T, \partial_T, \Upsilon_T, \Phi_T, \Gamma)$, where

- $\mathcal{M}$ *is a message space that is a set of strings (plaintexts),*
- $\mathcal{S}$ *is a signature space that is a set of strings (signatures),*
- Δ *is a random message space that is a set of strings,*
- $\mathcal{K} = \mathcal{K}_e \times \mathcal{K}_d$ *is a key space, such that* $\mathcal{K}_e$ *is the public key space and* $\mathcal{K}_d$ *is the private key space,*
- Λ *is a shadow key space,*
- $\Psi = \{U_i | 1 \leq i \leq n\}$ *is a set of* n *signers,*
- $\Re$ *is a set of requesters,*
- $\Omega_T : \Delta^n \rightarrow \mathcal{K}_e$ *is a poly-time distributed key generation protocol (secure computing protocol) used by all the signers* Ψ*. The private input of* U_i *is a random string* $\chi_i \in \Delta$*. The output of the protocol is the group public key* $K_e = \Omega_T(\chi_1, \chi_2, ..., \chi_n) \in \mathcal{K}_e$*. At the end of the protocol, the private output of signer* $U_i \in \Psi$ *is a secret shadow* $\theta_i \in \Lambda$*, such that the shadows* $\theta_i, 1 \leq i \leq n$*, form a* (t, n) *threshold secret sharing of* $K_d \in \mathcal{K}_d$*, where* K_d *is the corresponding private key of* K_e*.*
- $\partial_T : \mathcal{M} \times \Delta \times \mathcal{K}_e \times \Delta^t \rightarrow \mathcal{M}$ *is a poly-time blinding algorithm that on input a message* $m \in \mathcal{M}$*, a random blinding string* $\lambda \in \Delta$*, a public key* $K_e \in \mathcal{K}_e$ *and* $h(\delta_{P_i}) \in \Delta, 1 \leq i \leq t, 1 \leq P_1, P_t \leq n$ *and* $P_i < P_{i+1}$*, where* h *is a one-way hash function and* $\delta_{P_i} \in \Delta$*, constructs the blinded message* $m' = \partial_T(m, \lambda, K_e, h(\delta_{P_1}), h(\delta_{P_2}), ..., h(\delta_{P_t})) \in \mathcal{M}$*,*
- $\Upsilon_T : \mathcal{M} \times \mathcal{K}_e \times \Lambda^t \times \Delta^t \rightarrow \mathcal{S}$ *is a poly-time distributed signing protocol (secure computing protocol) used by any subset of* t *signers* $\{U_{P_i} | 1 \leq i \leq t, 1 \leq P_1, P_t \leq n$ *and* $P_i < P_{i+1}\}$*. The private input of* U_{P_i} *is the secret shadow* $\theta_{P_i} \in \Lambda$ *and the randomizing factor* $\delta_{P_i} \in \Delta$*. The public inputs consist of a blind message* $m' = \partial_T(m, \lambda, K_e, h(\delta_{P_1}), h(\delta_{P_2}), ..., h(\delta_{P_t})) \in \mathcal{M}$ *and the public key* $K_e \in \mathcal{K}_e$*. The output of the protocol is the blind signature* $s' = \Upsilon_T(m', K_e, \theta_{P_1}, \theta_{P_2}, ..., \theta_{P_t}, \delta_{P_1}, \delta_{P_2}, ..., \delta_{P_t}) \in \mathcal{S}$*.*
- $\Phi_T : \mathcal{S} \times \Delta \rightarrow \mathcal{S}$ *is a poly-time unblinding algorithm that on input a blind signature* $s' = \Upsilon_T(\partial_T(m, \lambda, K_e, h(\delta_{P_1}), h(\delta_{P_2}), ..., h(\delta_{P_t})), K_e, \theta_{P_1}, \theta_{P_2}, ..., \theta_{P_t}, \delta_{P_1}, \delta_{P_2}, ..., \delta_{P_t})) \in \mathcal{S}$ *and the random blinding string* λ*, extracts the signature* $s = \Phi_T(s', \lambda)$ *on* m*,*
- $\Gamma : \mathcal{M} \times \mathcal{S} \times \mathcal{K}_e \rightarrow \{true, false\}$ *is a poly-time verification algorithm that on input a message-signature pair* (m, s) *and a public key* $K_e \in \mathcal{K}_e$*, determines if* s *is a valid signature for message* m*,*

such that, we have the following:

1. *Before a requester* $R \in \Re$ *can request a blind threshold signature from any subset of* t *signers* $\Psi_t = \{U_{P_i} | 1 \leq i \leq t, 1 \leq P_1, P_t \leq n$ *and* $P_i < P_{i+1}\}$*, all the signers in* Ψ *have to apply* Ω_T *to construct a group public key* $K_e \in \mathcal{K}_e$*, where the corresponding group private key of* K_e *is* $K_d \in \mathcal{K}_d$*. At the end of* Ω_T*, each signer* $U_i \in \Psi$ *gets a secret shadow* $\theta_i \in \Lambda$*.*
2. *In a blind threshold signature generation, a requester* $R \in \Re$ *chooses a random string* $\lambda \in \Delta$ *and computes* $m' = \partial_T(m, \lambda, K_e, h(\delta_{P_1}), h(\delta_{P_2}), ..., h(\delta_{P_t}))$*, where* K_e *is* Ψ*'s group public key and* δ_{P_i} *is the randomizing factor chosen by* U_{P_i}*, for blinding a message* m *and submits* m' *to* $\Psi_t = \{U_{P_i} | 1 \leq i \leq t, 1 \leq P_1, P_t \leq n$ *and* $P_i < P_{i+1}\}$*.* Ψ_t *then apply the distributed signing protocol* Υ_T *to* m' *and send* R *the signing result* $s' = \Upsilon_T(m', K_e, \theta_{P_1}, \theta_{P_2}, ..., \theta_{P_t}, \delta_{P_1}, \delta_{P_2}, ..., \delta_{P_t}))$*, where* θ_{P_i} *is the secret shadow of* U_{P_i}*. After receiving* s'*,* R *extracts the signature* $s = \Phi_T(s', \lambda)$ *on the message* m*.*

3. *Anyone can verify if a message-signature pair (m, s) is valid for the group public key $K_e \in \mathcal{K}_e$ by the function Γ.*

4. *In a blind threshold signature generation, the signers' views ν and the message-signature pair (m, s) which is later made public are statistically independent.* □

The digital signature scheme with the restoration method can be defined similarly except the verification function Γ must be replaced by a restoration function Θ. To verify a signature $s \in \mathcal{S}$, one simply computes $m = \Theta(s, K_e)$ and checks if m has some redundancy information.

The notion of security for blind signature schemes was formally defined in [25] under the random oracle model.

Definition 2 ***(The "one-more forgery").*** For any fixed l, if an attacker A can compute, after l interactions with the signer, $l+1$ signatures with non-negligible probability, we say that it has performed an $(l, l+1)$-forgery. A "one-more forgery" is an $(l, l+1)$-forgery for some integer l. □

Definition 3 ***(Attacks).*** Two different attacks can be considered:

1. the sequential attack where the attacker can sequential interacts with the signer.

2. the parallel attack where the attacker can interact l times with the signer and send the challenges whenever he wants. □

Definition 4 A blind signature scheme $\mathcal{P} = (\mathcal{M}, \mathcal{S}, \Delta, \mathcal{K}, \Psi, \Re, \Omega, \partial, \Upsilon, \Phi, \Gamma)$ is unforgeable if no malicious adversary can do the one-more forgery with non-negligible probability in the random oracle model under the sequential or parallel attack. □

The notion of security for blind (t, n) threshold signature scheme $\mathcal{P}_T$ can be formally defined as follows.

Definition 5 A blind (t, n) threshold signature scheme is unforgeable, if no malicious adversary who corrupts at most $t-1$ signers can do one-more forgery with a honest signer in the random oracle model with non-negligible probability under the sequential or parallel attack. □

In order to prove unforgeability we use the concept of the simulatable adversary view [8, 10]. This means the adversary who sees all the information of the corrupted signers and the signature of m, could generate by itself all the other information produced by the protocol, except the secret information generated by the honest signer. In other words, the run of the protocol provides no useful information to the adversary other than the final signature on m. Indeed one can prove that if the underlying signature scheme $\mathcal{P}$ of a simulatable threshold signature scheme $\mathcal{P}_T$ is unforgeable then $\mathcal{P}_T$ is unforgeable [8, 10].

Definition 6 A blind (t, n) threshold signature scheme is simulatable if there exists a simulator SIM that on input the public key y, the public input m, the partial secret shadows provided by the $t-1$ corrupted signers and the signature s of m, can simulate the view of the adversary on an execution of the scheme that generate s as an output. □

3 The proposed scheme

In this section, we propose a blind threshold signature scheme based on the Okamoto-Schnorr blind signature scheme [25]. In a typical signing process of a blind threshold signature scheme, there are two kinds of participants, the signers and a requester. Before the requester can obtain a blind threshold signature from the signers, all the signers have to cooperate to distribute their secret shadows to other signers in advance. Then the requester requests a blind threshold signature from the signers. The proposed scheme consists of three phases: (1) the shadow distribution phase, (2) the signature generation phase and (3) the signature verification phase. The shadow distribution phase is performed only once among the signers and then they can use their secret shadows to sign messages. In the signature generation phase, a requester requests a blind threshold signature from the signers and the signers cooperate to issue the blind threshold signature to the requester. In the signature verification phase, anyone can use the group public key to verify if a blind threshold signature is valid.

Let U_i be the identification of signer i, n be the number of signers, t be the threshold value of the blind threshold signature scheme, m be the blind message to be signed, h be a secure one-way hashing function [20], p, q be two large prime numbers such that q divides $(p-1)$, and ξ, ξ' be two generators of Z_p^*. Let $x \equiv_p y$ denote $x = y \bmod p$. Let $g \equiv_p \xi^{(p-1)/q}$ and $h \equiv_p \xi'^{(p-1)/q}$. Let d_i be the secret key chosen by U_i. In a distributed environment, U_i can publish the corresponding public key e_i. Anyone can get e_i via some authentication service (e.g. the X.509 directory authentication service [30]). Using a secure public key signature scheme [6, 27], U_i can produce signatures of messages by his own secret key d_i. Anyone can verify these signatures by the corresponding public key e_i. Let $C(m, \gamma)$ denote a commitment to $m \in Z_p^*$ using the random string γ and $Cert_{U_i}(h(c))$ denote the signature on $h(c)$ signed by U_i.

3.1 The shadow distribution phase

Before a requester can request a blind threshold signature from the signers, all signers must cooperate to distribute their secret shadows to other signers. In the shadow distribution phase, each U_i, $1 \le i \le n$, carries out the following steps:

1. U_i chooses two secret keys $r_i, s_i \in Z_q$ and two secret polynomials $f_i(x) = \sum_{k=0}^{t-1} a_{i,k} x^k$ and $f_i'(x) = \sum_{k=0}^{t-1} a'_{i,k} x^k$ such that $a_{i,0} = r_i$,$a'_{i,0} = s_i$ and $a_{i,j}, a'_{i,j} \in Z_q, 1 \le j \le t-1$, computes $\Psi_{i,k} \equiv_p g^{-a_{i,k}}, \Psi'_{i,k} \equiv_p h^{-a'_{i,k}}, 0 \le k \le t-1$ and the signatures $Cert_{U_i}(h(\Psi_{i,k}))$ on $\Psi_{i,k}, Cert_{U_i}(h(\Psi'_{i,k}))$ on $\Psi'_{i,k}, 1 \le k \le t-1$,

the commitments $C_i = C(\Psi_{i,0}, \gamma_i), C'_i = C(\Psi'_{i,0}, \gamma'_i)$ and the signatures $Cert_{U_i}(h(C_i))$ on C_i and $Cert_{U_i}(h(C'_i))$ on C'_i and sends $(Cert_{U_i}(h(C_i)), C_i, Cert_{U_i}(h(C'_i)), C'_i, (\Psi_{i,k}, \Psi'_{i,k}, Cert_{U_i}(h(\Psi_{i,k})), Cert_{U_i}(h(\Psi'_{i,k}))$, $1 \le k \le t-1$) to U_j, $1 \le j \le n$, $j \ne i$.

2. Upon receiving $(Cert_{U_j}(h(C_j)), C_j, Cert_{U_j}(h(C'_j)), C'_j, (\Psi_{j,k}, \Psi'_{j,k}, Cert_{U_j}(h(\Psi_{j,k})), Cert_{U_j}(h(\Psi'_{j,k}))), 1 \le j \le n, j \ne i, 1 \le k \le t-1$) from all other signers, U_i verifies if all $Cert_{U_j}(h(C_j)), Cert_{U_j}(h(C'_j)), Cert_{U_j}(h(\Psi_{j,k}))$, $Cert_{U_i}(h(\Psi'_{i,k}))$ are valid. If valid, he opens C_i, C'_i sends $\delta_{i,j} \equiv_q f_i(x_j), \delta'_{i,j} \equiv_q f'_i(x_j)$, where x_j is a unique public number for U_j, and a signature $Cert_{U_i}(h(\delta_{i,j}))$ on $\delta_{i,j}, Cert_{U_i}(h(\delta'_{i,j}))$ on $\delta'_{i,j}$ secretly to every U_j, $1 \le j \le n$, $j \ne i$. Otherwise, he publishes the invalid signatures and stops.

3. When U_i receives all $\delta_{j,i}, \delta'_{j,i}$, $Cert_{U_j}(h(\delta_{j,i}))$,$Cert_{U_j}(h(\delta'_{j,i}))$, $1 \le j \le n$, $j \ne i$, from other signers, he verifies if the shares $\delta_{j,i}, \delta'_{j,i}$ received from U_j is consistent with the certified values $\Psi_{j,l}, \Psi'_{j,l}$, $0 \le l \le t-1$, by checking whether $g^{\delta_{j,i}} \equiv_p \prod_{l=0}^{t-1}(\Psi_{j,l})^{x_i{}^l}$ and $h^{\delta'_{j,i}} \equiv_p \prod_{l=0}^{t-1}(\Psi'_{j,l})^{x_i{}^l}$. If it fails, U_i broadcasts that an error has been found, publishes $\delta_{j,i}, Cert_{U_j}(h(\delta_{j,i}))$ or $\delta'_{j,i}, Cert_{U_j}(h(\delta'_{j,i}))$and the identification of U_j, and then stops. Otherwise, U_i computes the signature $Cert_{U_i}(h(y))$ on the group public key $y \equiv_p \prod_{l=1}^{n} y_l \equiv_p \prod_{l=1}^{n}\Psi_{l,0}\prod_{l=1}^{n}\Psi'_{l,0}$ and the signatures $Cert_{U_i}(h(\Phi_{j,i}))$ on $\Phi_{j,i} \equiv_p g^{\delta_{j,i}}$ and $Cert_{U_i}(h(\Phi'_{j,i}))$ on $\Phi'_{j,i} \equiv_p h^{\delta'_{j,i}}$, $1 \le j \le n$. He then sends $(Cert_{U_i}(h(y)), (\Phi_{j,i}, \Phi'_{j,i}, Cert_{U_i}(h(\Phi_{j,i}), Cert_{U_i}(h(\Phi'_{j,i})), 1 \le j \le n))$ to all other signers.

4. Upon receiving all $((Cert_{U_j}(h(y)), 1 \le j \le n, j \ne i), (\Phi_{j,i}, \Phi'_{j,i}, Cert_{U_j}(h(\Phi_{j,i}), Cert_{U_j}(h(\Phi'_{j,i})), 1 \le l \le n, 1 \le j \le n, j \ne i))$, U_i verifies if all $((Cert_{U_j}(h(y)), 1 \le j \le n, j \ne i), Cert_{U_j}(h(\Phi_{j,i}), Cert_{U_j}(h(\Phi'_{j,i})), 1 \le l \le n, 1 \le j \le n, j \ne i))$ are valid. If yes, the shadow keys corresponding to the group secret keys $s \equiv_q \sum_{j=1}^{n} s_j$, $r \equiv_q \sum_{j=1}^{n} r_j$ have been securely and correctly distributed. The group public key $y \equiv_p \prod_{l=1}^{n}\Psi_{l,0}\Psi'_{l,0}$, all signers' public keys $\Psi_{l,0}, \Psi'_{l,0}, 1 \le l \le n$, and all public shadows $\Phi_{l,j} \equiv_p g^{\delta_{l,j}}, \Phi'_{l,j} \equiv_p g^{\delta'_{l,j}}$, $1 \le l, j \le n$, can then be published by each signer. Otherwise, U_i publishes the invalid signatures and stops.

3.2 The signature generation phase

Without loss of generality, we assume that t out of n signers are U_i, $1 \le i \le t$. The t signers perform the following steps during the signature generation phase.

1. Each U_i randomly chooses two random numbers $t_i, u_i \in Z_q$, computes $a_i \equiv_p g^{t_i}h^{u_i}$ and sends a_i to the requester.

2. After receiving all $a_i, 1 \le i \le t$, the requester chooses three random numbers γ, β and $\delta \in Z_q$, computes $a \equiv_p \prod_{i=1}^{t} a_i, \alpha \equiv_p g^{\beta}h^{\gamma}y^{\delta}a, \varepsilon \equiv_p H(m, \alpha)$ and $e \equiv_q \varepsilon - \delta$ and sends e to all $U_i, 1 \le i \le t$.

3. Upon receiving e, each U_i computes $R_i \equiv_q e(r_i + \sum_{j=t+1}^{n} f_j(x_i)(\prod_{k=1,k\ne i}^{t}(\frac{-x_k}{x_i-x_k}))) + t_i, S_i \equiv_q e(s_i + \sum_{j=t+1}^{n} f'_j(x_i)(\prod_{k=1,k\ne i}^{t}(\frac{-x_k}{x_i-x_k}))) + u_i$ and sends S_i and R_i back to the requester.

4. After receiving all S_i and R_i, the requester checks if

$$g^{R_i}h^{S_i}y^e \equiv_p a_i(\prod_{j=t+1}^{n}(\Psi_{j,i}))^{(\prod_{k=1,k\ne i}^{t}(\frac{-x_k}{x_i-x_k}))e}(\prod_{j=t+1}^{n}(\Psi'_{j,i}))^{(\prod_{k=1,k\ne i}^{t}(\frac{-x_k}{x_i-x_k}))e}, 1 \le i \le t.$$

If any of the S_i and R_i is not valid, he has to ask the corresponding signer to send it again. Otherwise, he computes $\rho \equiv_q \beta + \sum_{i=1}^{t} R_i, \sigma \equiv_q \gamma + \sum_{i=1}^{t} S_i$. The blind threshold signature of m is (α, ρ, σ).

3.3 The signature verification phase

To verify the blind threshold signature (α, ρ, σ) on the message m, one simply checks if $\alpha \equiv_p g^{\rho}h^{\sigma}y^{\varepsilon}$.

4 Discussion

We discuss the correctness, security, performance and extensions of our blind threshold signature scheme in this section.

4.1 Correctness

To prevent a signer from sending an invalid partial signature to the requester, the partial signature must be checked in step 4 of the signature generation phase. The following lemma ensures the *correctness of partial signatures*.

Lemma 1. *The partial signature (R_i, S_i) is valid if U_i is honest.*

Proof. By our scheme, we have

$$\begin{aligned}
& g^{R_i}h^{S_i}y_i^e \\
\equiv_p\ & g^{e(r_i+\sum_{j=t+1}^{n} f_j(x_i)(\prod_{k=1,k\ne i}^{t}(\frac{-x_k}{x_i-x_k})))+t_i} \\
& h^{e(s_i+\sum_{j=t+1}^{n} f'_j(x_i)(\prod_{k=1,k\ne i}^{t}(\frac{-x_k}{x_i-x_k})))+u_i}y_i^e \\
\equiv_p\ & g^{er_i}g^{e\sum_{j=t+1}^{n} f_j(x_i)(\prod_{k=1,k\ne i}^{t}(\frac{-x_k}{x_i-x_k}))}g^{t_i}h^{es_i} \\
& h^{e\sum_{j=t+1}^{n} f'_j(x_i)(\prod_{k=1,k\ne i}^{t}(\frac{-x_k}{x_i-x_k}))}h^{u_i}(g^{-r_i}h^{-s_i})^e \\
\equiv_p\ & g^{e\sum_{j=t+1}^{n} f_j(x_i)(\prod_{k=1,k\ne i}^{t}(\frac{-x_k}{x_i-x_k}))}g^{t_i} \\
& h^{e\sum_{j=t+1}^{n} f'_j(x_i)(\prod_{k=1,k\ne i}^{t}(\frac{-x_k}{x_i-x_k}))}h^{u_i}
\end{aligned}$$

$\equiv_p a_i(\prod_{j=t+1}^{n}(\Psi_{j,i}))^{(\prod_{k=1,k\neq i}^{t}(\frac{-x_k}{x_i-x_k}))}$

$(\prod_{j=t+1}^{n}(\Psi'_{j,i}))^{(\prod_{k=1,k\neq i}^{t}(\frac{-x_k}{x_i-x_k}))}$. □

After the signature generation phase, the blind threshold signatures can be verified by the group public key in the signature verification phase. Let ν denote the signers' complete views of an execution in the signature generation phase and let $(m, (\alpha, \rho, \sigma))$ denote the message-signature pair generated in that execution. Theorem 2 ensures the correctness of the scheme.

Theorem 2. *The 3-tuple (α, ρ, σ) is a valid blind threshold signature on the message m.*

Proof. The validity of the blind threshold signature (α, ρ, σ) on the message m can easily be established as follows.

$g^{\rho}h^{\sigma}y^{\varepsilon}$

$\equiv_p g^{\beta+\sum_{i=1}^{t}R_i}h^{\gamma+\sum_{i=1}^{t}S_i}y^{\varepsilon}$

$\equiv_p g^{\beta+\sum_{i=1}^{t}R_i}h^{\gamma+\sum_{i=1}^{t}S_i}y^{e+\delta}$

$\equiv_p g^{\beta}h^{\gamma}g^{\sum_{i=1}^{t}t_i+e\sum_{i=1}^{t}r_i}h^{\sum_{i=1}^{t}u_i+e\sum_{i=1}^{t}s_i}$

$g^{e\sum_{i=1}^{t}\sum_{j=t+1}^{n}f_j(x_i)(\prod_{k=1,k\neq i}^{t}(\frac{-x_k}{x_i-x_k}))}$

$h^{e\sum_{i=1}^{t}\sum_{j=t+1}^{n}f'_j(x_i)(\prod_{k=1,k\neq i}^{t}(\frac{-x_k}{x_i-x_k}))}y^{e+\delta}$

$\equiv_p ag^{\beta}h^{\gamma}g^{e(\sum_{i=1}^{t}r_i+\sum_{i=t+1}^{n}r_i)}h^{e(\sum_{i=1}^{t}s_i+\sum_{i=t+1}^{n}s_i)}$

$y^{e+\delta}$

$\equiv_p ag^{\beta}h^{\gamma}y^{-e}y^{e+\delta}$

$\equiv_p ag^{\beta}h^{\gamma}y^{\delta}$

$\equiv_p \alpha$. □

4.2 Security analysis

In the shadow distribution phase, since $\Psi_{i,0}$ and $\Psi'_{i,0}$ are committed using γ_i and γ'_i and after U_i receiving all other commitments $C_j = C(\Psi_{i,0}, \gamma_j)$ and $C'_j = C(\Psi'_{i,0}, \gamma'_j)$, $1 \leq j \leq n$, $j \neq i$, he opens the commitments, if U_i chooses his secret keys r_i and s_i at random then the distributions of the group secret keys $s \equiv_q \sum_{j=1}^{n} s_j$ and $r \equiv_q \sum_{j=1}^{n} r_j$ are both polynomially indistinguishable from the uniform distribution. Given the secret information of a group of $l < t$ members, Lemma 3 ensures that the threshold cryptosystem constructed in the shadow distribution phase will not disclose any extra information about the group secret keys $s \equiv_q \sum_{j=1}^{n} s_j$ and $r \equiv_q \sum_{j=1}^{n} r_j$.

Lemma 3. Given a group of $\sigma < t$ members $G = \{p_i | p_i \in [1, n], 1 \leq i \leq \sigma\}$ and the set of shares $\{\delta_{j,i}, \delta'_{j,i} | 1 \leq j \leq n, i \in G\}$. For any fixed j, $1 \leq j \leq n$, it takes polynomial time on $|p|$ to generate two random sets $\{g^{\widehat{a_{j,k}}} | 1 \leq k \leq t-1\}$ and $\{h^{\widehat{a'_{j,k}}} | 1 \leq k \leq t-1\}$ satisfying $g^{\delta_{j,i}} \equiv_p \prod_{k=0}^{t-1}(g^{\widehat{a_{j,k}}})^{x_i{}^k}$ and $h^{\delta'_{j,i}} \equiv_p \prod_{k=0}^{t-1}(h^{\widehat{a'_{j,k}}})^{x_i{}^k}$ for $i \in G$.

Proof. In step 3 of the shadow distribution phase, after U_i has received all $\delta_{j,i}$, he verifies if the share $\delta_{j,i}$ received from U_j is consistent with the certified values $\Psi_{j,l}$, $1 \leq l \leq t-1$, by checking if $g^{\delta_{j,i}} \equiv_p \prod_{l=0}^{t-1}(\Psi_{j,l})^{x_i{}^l}$. Therefore

$$g^{\delta_{j,i}} \equiv_p \prod_{l=0}^{t-1}(g^{a_{j,l}})^{x_i{}^l} \equiv_p g^{\sum_{l=0}^{t-1} a_{j,l} * x_i{}^l}. \quad (1)$$

Since $g \equiv_p \xi^{(p-1)/q}$ and ξ is a generator of Z_p^*, g generates a cyclic subgroup S_q of Z_p^* with $|S_q| = q$. From (1), we have

$$\delta_{j,i} \equiv_q \sum_{l=0}^{t-1} a_{j,l} * x_i{}^l \quad (2)$$

From (2), we know that given a fixed index j, the shares $\delta_{j,i}$, $i \in G$, will use the same variables $\widehat{a_{j,k}}$, $0 \leq k \leq t-1$, as follows:

$$\delta_{j,i} \equiv_q \sum_{k=0}^{t-1} \widehat{a_{j,k}} * x_i{}^k. \quad (3)$$

Given a fixed index j, we can get at most σ linear equations with t variables as follows:

$$\delta_{j,i} \equiv_q \sum_{k=0}^{t-1} \widehat{a_{j,k}} * x_i{}^k (i \in G). \quad (4)$$

Since the linear equations have at least one solution $\widehat{a_{j,k}} = a_{j,k}$, $0 \leq k \leq t-1$, we can solve the linear equations (4) and get a random solution $\widehat{a_{j,k}}$, $1 \leq k \leq t-1$, by assigning random values to all free variables. From (4), it is clear that $g^{\delta_{j,i}} \equiv_p g^{\sum_{k=0}^{t-1} \widehat{a_{j,k}} * x_i{}^k} \equiv_p \prod_{k=0}^{t-1}(g^{\widehat{a_{j,k}}})^{x_i{}^k}$.

Similar to the above proof, we can get a random solution $\widehat{a'_{j,k}}$, $1 \leq k \leq t-1$, such that, $h^{\delta'_{j,i}} \equiv_p h^{\sum_{k=0}^{t-1} \widehat{a'_{j,k}} * x_i{}^k} \equiv_p \prod_{k=0}^{t-1}(h^{\widehat{a'_{j,k}}})^{x_i{}^k}$. □

Let ν denote the signers' complete views of an execution in the signature generation phase and let $(m, (\alpha, \rho, \sigma))$ denote the message-signature pair generated in that execution. Theorem 4 ensures the blindness of our proposed scheme.

Theorem 4. *The threshold signature scheme proposed in Section 4 is blind.*

Proof. For proving the blindness of the scheme, we show that given any view ν and any valid message-signature pair $(m, (\alpha, \rho, \sigma))$, there exists a unique triple of blinding factors β, γ and δ. Since the requester chooses the blinding factors β, γ and δ randomly, the blindness of the signature scheme follows.

Given a valid message-signature pair $(m, (\alpha, \rho, \sigma))$ and a view ν, the following equations must hold for β, γ and δ. Without loss of generality, assume that the blind signature (α, ρ, σ) has been generated by t signers U_i, $1 \leq i \leq t$, with the view ν consisting of $R_i \equiv_q e(r_i + \sum_{j=t+1}^{n} f_j(x_i)(\prod_{k=1,k\neq i}^{t} (\frac{-x_k}{x_i-x_k}))) + t_i, S_i \equiv_q e(s_i + \sum_{j=t+1}^{n} f'_j(x_i)(\prod_{k=1,k\neq i}^{t} (\frac{-x_k}{x_i-x_k}))) + u_i, t_i$ and $u_i, 1 \leq i \leq t$ and e.

$$\rho \equiv_q \beta + \sum_{i=1}^{t} R_i \quad (5)$$

$$\sigma \equiv_q r + \sum_{i=1}^{t} S_i \quad (6)$$

$$\varepsilon \equiv_q e + \delta \quad (7)$$

By equations (5), (6) and (7), the unique solution for β, γ and δ is:

$$\beta \equiv_q \rho - \sum_{i=1}^{t} R_i \quad (8)$$

$$\gamma \equiv_q \sigma - \sum_{i=1}^{t} S_i \quad (9)$$

$$\delta \equiv_q \varepsilon - \sigma \quad (10)$$

□

Our proposed blind signature scheme is based on a provably secure blind signature scheme under the random oracle model [25].

Theorem 5. Consider the Okamato-Schnorr blind signature scheme in the random oracle model. A "one-more forgery", even under a parallel attack, is equivalent to the discrete logarithm problem in a subgroup. [25] □

Since the Okamato-Schnorr blind signature scheme is unforgeable in the random oracle model, if our proposed blind threshold signature scheme is simulatable, our proposed scheme is unforgeable.

Let *Threshold_gen* denote the protocol in the signature generation phase. Without loss of generality, we assume that the adversary has corrupted $t-1$ signers U_i, $1 \le i \le t-1$, and the requester with the view consisting of $m, y, (r_i, s_i, 1 \le i \le t-1), (\delta_{i,j}, \delta'_{i,j}, 1 \le i \le t-1, 1 \le j \le n)$. To prove the unforgeability of our proposed scheme, we now construct a simulator SIM as follows. The simulator SIM is described as a two phase protocol. The first phase computes all the necessary information, and in the second phase it carries out the communication with the adversary in accordance with *Threshold_gen*.

Simulator SIM
SIM_Computation$(m, y, (r_i, s_i, 1 \le i \le t-1), (\delta_{i,j}, \delta'_{i,j}, 1 \le i \le t-1, 1 \le j \le n), (\alpha, \rho, \sigma))$

1. Randomly choose $\widetilde{t_i}$ and $\widetilde{u_i} \in Z_q, 1 \le i \le t-1$.
2. Randomly choose $\widetilde{\gamma}, \widetilde{\beta}$ and $\widetilde{\delta} \in Z_q$ and compute $\widetilde{e} \equiv_q \varepsilon - \widetilde{\delta}$.
3. Compute $\widetilde{R_i} \equiv_q \widetilde{e}(r_i + \sum_{j=t+1}^{n} f_j(x_i)(\prod_{k=1,k\neq i}^{t}(\frac{-x_k}{x_i-x_k}))) + \widetilde{t_i}, 1 \le i \le t-1$.
4. Compute $\widetilde{S_i} \equiv_q \widetilde{e}(s_i + \sum_{j=t+1}^{n} f'_j(x_i)(\prod_{k=1,k\neq i}^{t}(\frac{-x_k}{x_i-x_k}))) + \widetilde{u_i}, 1 \le i \le t-1$.
5. Compute $\widetilde{R_t} \equiv_q \rho - \widetilde{\beta} - \sum_{i=1}^{t-1} \widetilde{R_i}$.
6. Compute $\widetilde{S_t} \equiv_q \sigma - \widetilde{\gamma} - \sum_{i=1}^{t-1} \widetilde{S_i}$.

end of SIM_Computation.

SIM_Conversation

Comment: In each of the following steps, we describe the information which SIM gives to the adversary. Each of these steps relates to the same numbered step in protocol *Threshold_gen*.

1. The $2(t-1)$ random numbers $\widetilde{t_i}$ and $\widetilde{u_i} \in Z_q, 1 \le i \le t-1$.
2. The three blinding factors $\widetilde{\gamma}$, $\widetilde{\delta}$ and $\widetilde{\beta}$ and the blind message $\widetilde{e}$.
3. The $2t$ blind partial signatures $\widetilde{R_i} \equiv_q \widetilde{e}(r_i + \sum_{j=t+1}^{n} f_j(x_i)(\prod_{k=1,k\neq i}^{t}(\frac{-x_k}{x_i-x_k}))) + \widetilde{t_i}, 1 \le i \le t-1$, $\widetilde{R_t} \equiv_q \rho - \widetilde{\beta} - \sum_{i=1}^{t-1} \widetilde{R_i}$, $\widetilde{S_i} \equiv_q \widetilde{e}(s_i + \sum_{j=t+1}^{n} f'_j(x_i)(\prod_{k=1,k\neq i}^{t}(\frac{-x_k}{x_i-x_k}))) + \widetilde{u_i}, 1 \le i \le t-1$ and $\widetilde{S_t} \equiv_q \sigma - \widetilde{\gamma} - \sum_{i=1}^{t-1} \widetilde{S_i}$.
4. Do nothing.

end of SIM_Conversation.
end of SIM.

Let $View_A(Threshold\_gen(m, y, (r_i, s_i, 1 \le i \le t-1), (\delta_{i,j}, \delta'_{i,j}, 1 \le i \le t-1, 1 \le j \le n), (\alpha, \rho, \sigma)))$ be all the information of the corrupted signers and the requester in the signature generation phase and $SIM(m, y, (r_i, s_i, 1 \le i \le t-1), (\delta_{i,j}, \delta'_{i,j}, 1 \le i \le t-1, 1 \le j \le n), (\alpha, \rho, \sigma))$ be the information constructed by the simulator SIM with $(m, y, (r_i, s_i, 1 \le i \le t-1), (\delta_{i,j}, \delta'_{i,j}, 1 \le i \le t-1, 1 \le j \le n), (\alpha, \rho, \sigma))$ as input. Theorem 6 ensures that *Threshold_gen* in Section 3.2 is simulatable.

Theorem 6. $View_A(Threshold\_gen(m, y, (r_i, s_i, 1 \le i \le t-1), (\delta_{i,j}, \delta'_{i,j}, 1 \le i \le t-1, 1 \le j \le n), (\alpha, \rho, \sigma)))$ is computationally indistinguishable from $SIM(m, y, (r_i, s_i, 1 \le i \le t-1), (\delta_{i,j}, \delta'_{i,j}, 1 \le i \le t-1, 1 \le j \le n), (\alpha, \rho, \sigma))$.

Proof. We shall analyze the information generated by *Threshold_gen* and SIM in each step.

1. Both *Threshold_gen* and SIM choose $2(t-1)$ random numbers. Thus generate the same probability distribution for the sets of size $2(t-1)$.
2. *Threshold_gen* randomly chooses three blinding factors γ, β and $\delta \in Z_q$ and SIM also randomly chooses three blinding factors $\widetilde{\gamma}, \widetilde{\beta}$ and $\widetilde{\delta} \in Z_q$. These three probability distributions are the same. *Threshold_gen* computes the blind message $e \equiv_q \varepsilon - \delta$ and SIM computes the blind message $\widetilde{e} \equiv_q \varepsilon - \widetilde{\delta}$. These two blind messages are both blinded with random blind factors δ or $\widetilde{\delta}$. So these two probability distributions are the same.
3. *Threshold_gen* generates t blind partial signatures $R_i \equiv_q e(r_i + \sum_{j=t+1}^{n} f_j(x_i)(\prod_{k=1,k\neq i}^{t}(\frac{-x_k}{x_i-x_k}))) + t_i, 1 \le i \le t$, which consist of the blind message e, the partial secrets $r_i + \sum_{j=t+1}^{n} f_j(x_i)(\prod_{k=1,k\neq i}^{t}(\frac{-x_k}{x_i-x_k})), 1 \le i \le t$, and the random numbers $t_i, 1 \le i \le t$. SIM also generates t blind partial signatures $\widetilde{R_i} \equiv_q \widetilde{e}(r_i + \sum_{j=t+1}^{n} f_j(x_i)(\prod_{k=1,k\neq i}^{t}(\frac{-x_k}{x_i-x_k}))) + t_i, 1 \le i \le t-1$, which consist of the blind message $\widetilde{e}$, the partial secrets

Table 1: Cost of the signature generation phase in the blind threshold signature scheme and that in the underlying blind signature scheme.

	The requester			
	EXP	INV	MUL	ADD
Scheme 1	3	0	$t+2$	$2t+1$
Scheme 1*	3	0	3	3

	The signer or U_i			
	EXP	INV	MUL	ADD
Scheme 1	2	0	$2n-1$	$2(n-t+1)$
Scheme 1*	2	0	3	2

where
EXP = the no. of modulo exponentiations,
INV = the no. of modulo inversions (divisions),
MUL = the no. of modulo multiplications,
ADD = the no. of modulo additions.

$r_i + \sum_{j=t+1}^{n} f_j(x_i)(\prod_{k=1,k\neq i}^{t} (\frac{-x_k}{x_i-x_k})), 1 \leq i \leq t-1$, and the random numbers $\widetilde{t_i}, 1 \leq i \leq t-1$, and $\widetilde{R_t} \equiv_q \rho - \widetilde{\beta} - \sum_{i=1}^{t-1} \widetilde{R_i}$. Since the blind messages $\widetilde{e}$ and e are in the same probability distribution, the partial signatures R_i and $\widetilde{R_i}, 1 \leq i \leq t-1$, are in the same probability distribution. In step 3, we can know that R_t and $\widetilde{R_t}$ are in the same probability distribution since β and $\widetilde{\beta}$ are in the same probability distribution. Similarly, we can show that the partial signatures S_i and $\widetilde{S_i}, 1 \leq i \leq t$, are in the same probability distribution.

This completes the proof of Theorem 6. □

Since the underlying blind signature scheme is unforgeable and our proposed threshold signature scheme is simulatable, the security of the proposed threshold signature scheme is equivalent to the discrete logarithm problem.

4.3 Performance analysis

In this subsection we give an analysis of the computational cost required to compute blind (t,n) threshold signatures in our scheme. We use as a measure the number of modular exponentiations and that of modular inverses required by a single player during the execution of our signature generation protocol. Table 1 illustrates the comparison of blind threshold signature scheme and its underlying blind signature scheme. In this table, Scheme 1 denotes the blind threshold signature scheme in Section 4 and Scheme 1* denotes its corresponding underlying blind signature scheme. For reducing the computational cost needed by each signer, the value $-x_k/(x_i - x_k)$, $1 \leq k \leq n$ and $k \neq i$, in Step 3 of the signature generation phase can be computed off-line. In this case, each signer needs to compute only 2 modular exponentiation in our scheme which is the same as the underlying blind signature schemes. Compared with the underlying blind signature scheme, the extra cost for signing a blind threshold signature is to compute $\sum_{j=t+1}^{n} f_j(x_i)(\prod_{k=1,k\neq i}^{t} (\frac{-x_k}{x_i-x_k}))$ and $\sum_{j=t+1}^{n} f'_j(x_i)(\prod_{k=1,k\neq i}^{t} (\frac{-x_k}{x_i-x_k}))$ in Step 3 which contains $2(n-2)$ modular multiplications and $2(n-t)$ additions. For reducing the computational cost needed by the requester, the partial signature verification in step 4 would not be done except the final threshold signature can not pass the verification equation in the signature verification phase. In this approach, the requester only needs to perform 3 modular exponentiations in Step 2 of the signature generation phase which is the same as the underlying blind signature scheme. Since the blind threshold verification function of our scheme is the same as that of the underlying blind signature scheme, the verification cost of our blind threshold signature is the same as that of the underlying blind signature. Compared with the underlying blind signature scheme, the extra cost for requesting a blind threshold signature in our scheme proposed in Section 4 is to compute $\prod_{i=1}^{t} a_i$, $\sum_{i=1}^{t} R_i$ and $\sum_{i=1}^{t} S_i$ which contains $t-1$ modular multiplications and $2(t-1)$ modular additions. In our scheme, the size of the threshold signature is the same as that of an individual signature and the verification process of a threshold signature is equivalent to that of an individual signature. Thus, our proposed scheme is optimal with respect to the threshold signature size and the verification process.

In [8], three robust threshold signature protocols, namely, DSS-Thresh-Sig-1, DSS-Thresh-Sig-2 and DSS-Thresh-Sig-3, are proposed. One approach to generate blind threshold signatures is to take robust threshold signature schemes [8] and turn them into blind signature schemes. The advantage of this approach is that it is quite robust and can deal with the situation where there are many cheaters. However, in DSS-Thresh-Sig-1, $2t+3$ modular exponentiations are required for each signer to generate a threshold signature and it is even worse for DSS-Thresh-Sig-2 and DSS-Thresh-Sig-3 which requires $O(nt)$ modular exponentiations. It is clear that this approach is quite inefficient compared to our proposed scheme.

5 Conclusion

We proposes an efficient and provably secure blind threshold signature scheme based on discrete logarithm. In our scheme, the size of a blind threshold signature is the same as that of an individual blind signature and the signature verification process is equivalent to that of an individual signature. Our proposed scheme is the first scheme, such that, its security is proved as equivalent as the discrete logarithm problem. Our proposed scheme can be easily applied to current efficient single-authority e-cash schemes for distributing the power of a single authority without changing the underlying structure and degrading the overall performance.

References

[1] M. Abe and E. Fujisaki, "How to date blind signatures," Proc. of AisaCrypt'96, LNCS 1163, pp. 244-251, Springer-Verlag, 1996.

[2] J. Camenisch, J. Pivereau and M. Stadler, "Blind signatures based on the discrete logarithm problem," Proc. of EuroCrypt'94, LNCS 950, pp. 428-432, Springer-Verlag, 1995.
[3] D. Chaum, "Untraceable electronic mail, return addresses, and digital pseudonyms," *Commu. of the ACM*, 24(2) (1981) pp. 84-88.
[4] D. Chaum, "Blind signatures for untraceable payments," Proc. of Crypt'82, pp. 199-203, Plenum, NY, 1983.
[5] D. Chaum, "Privacy protected payments: unconditional payer and/or payee untraceability," *In Smartcard 2000*, North Holland, 1988.
[6] T. ElGamal, "A public key cryptosystem and a signature scheme based on discrete logarithm," *IEEE Trans. on Information Theory*, IT-31(4) (1985), pp. 469-472.
[7] N. Ferguson, "Single term off-line coins," Proc. of EuroCrypt'93, LNCS 765, pp. 318-328, Springer-Verlag, 1993.
[8] R. Gennaro, S. Jarecki, H. Krawczyk and T. Rabin, "Robust threshold DSS signatures," Proc. of EuroCrypt '96, LNCS 1070, pp. 354-371. Springer Verlag, 1996.
[9] R. Gennaro, S. Jarecki, H. Krawczyk and T. Rabin, "On the (In) Security of Composed VSS Protocols," the rump session at Crypto'98, 1998.
[10] S. Goldwasser, S. Micali and C. Rackoff, "The knowledge complexity of interactive proof-systems," *SIAM Journal Computing*, 18 (1) (1989), pp. 186-208.
[11] L. Harn, "Group-oriented (t, n) threshold digital signature scheme and digital multisignature," *IEE Proc. Compu. Digit. Tech.*, 141(5) (1994), pp. 307- 313.
[12] P. Horster, M. Michels and H. Petersen, "Meta-message recovery and meta-blind signature schemes based on the discrete logarithm problem and their applications," Proc. of AisaCrypt'94, LNCS 917, Springer-Verlag, 1994, pp. 224-237.
[13] P. Horster, M. Michels and H. Petersen, "Comment on Cryptanalysis of the blind signatures based on the discrete logarithm problem," *Electronics Letters*, 31(21) (1995), pp.1827-1827.
[14] W. Juang and C. Lei, "A collision free secret ballot protocol for computerized general elections," *Computers & Security*, 15 (4) (1996) 339-348.
[15] W. Juang and C. Lei, "A secure and practical electronic voting scheme for real world environments," *IEICE Trans. on Fundamentals*, E80-A(1) (1997), pp. 64-71.
[16] W. Juang and C. Lei, "Blind threshold signatures based on discrete logarithm," *Proc. of Second Asian Computing Science Conference on Programming, Concurrency and Parallelism, Networking and Security*, LNCS 1179, pp. 172 -181, Springer-Verlag, 1996.
[17] W. Juang and C. Lei, "Partially blind threshold signatures based on discrete logarithm", Computer Communications, 22(1), pp. 73-86, 1999.
[18] R. C. Merkle, "One way hash functions and DES," Proc. of Crypt'89, LNCS 435, pp. 428-446, Springer-Verlag, 1990.
[19] S. Micali and P. Rogaway, "Secure computation," *Proc. of Crypt'91*, LNCS 576, pp. 392-404, Springer-Verlag, 1992.
[20] NIST FIPS PUB 180, "Secure hash standard," National Institute of Standards and Technology, U. S. Department of Commerce, DRAFT, 1993.
[21] NIST FIPS PUB XX, "Digital signature standard," National Institute of Standards and Technology, U.S. Department of Commerce, DRAFT, 1993.
[22] K. Nyberg and R. A. Rueppel, "Message recovery for signature schemes based on the discrete logarithm problem," *Proc. of EuroCrypt'94*, LNCS 950, pp. 182-193, Springer-Verlag, 1995.
[23] T. Okamoto, "A digital multisignature scheme using bijective public-key cryptosystems," *ACM Trans. Computer Systems*, 6(8), pp. 432-441, 1988.
[24] S. Pohlig and M. E. Hellman, "An improved algorithm for computing logarithms over GF(p) and its cryptographic significance," *IEEE Trans. on Information Theory*, IT-24, pp. 106-110, 1978.
[25] D. Pointcheval and J. Stern, "Provably secure blind signature schemes," Proc. of AisaCrypt'96, LNCS 1163, pp. 252-265, Springer-Verlag, 1996.
[26] D. Pointcheval and J. Stern, "New blind signatures equivalent to factorization," Proc. of the 4th ACM Conference on Computer and Communications Security, Zurich, Switzerland, 1997.
[27] R. L. Rivest, A. Shamir and L. Adelman, "A method for obtaining digital signatures and public key cryptosystem," *Commun. ACM*, 21(2), 1978, pp. 120-126.
[28] R. L. Rivest, "The MD5 message-digest algorithm," RFC 1321, Internet Activities Board, Internet Privacy Task Force, 1992.
[29] K. Sako, "Electronic voting scheme allowing open objection to the tally," *IEICE Trans. on fundamentals*, E77-A(1) (1994), pp. 24-30.
[30] W. Stallings, "Network and internetwork security," Prentice Hall International, 1995, pp. 333-340.