Fair Blind Threshold Signatures Based on Discrete Logarithm

Wen-Shenq Juang and Chin-Laung Lei*

Department of Electrical Engineering, Rm. 343
National Taiwan University
Taipei, Taiwan, R.O.C.

Abstract

In this paper, we propose a group-oriented fair blind (t, n) threshold signature scheme based on the discrete logarithm problem. By the scheme, any t out of n signers in a group can represent the group to sign fair blind threshold signatures, which can be used in anonymous e-cash systems. Since blind signature schemes provide perfect unlinkability, such e-cash systems can be misused by criminals, e.g. to safely obtain a ransom or to launder money. Our scheme allows the judge (or the government) to deliver information allowing anyone of the t signers to link his view of the protocol and the message-signature pair. In our scheme, the size of a fair blind threshold signature is the same as that of an individual fair blind signature and the signature verification process is simplified by means of a group public key. The security of our scheme relies on the difficulty of computing discrete logarithm.

Keywords: Fair Blind Signatures, Threshold Signatures, Discrete Logarithm, Privacy and Security, Secure E-Cash Systems.

1 Introduction

The concept of blind signature was introduced by Chaum [1]. It allows a requester to obtain signatures on the messages he provides to the signer without revealing these messages. A distinguishing property required by a typical blind signature scheme [1, 2, 3, 4] 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 secure electronic payment schemes [1, 5, 6, 7] protecting customers' anonymity, and secure voting schemes [8, 9, 10] preserving voters' privacy. In a distributed environment, the signed blind messages can be regarded as a fixed amount of electronic money in secure electronic

*Corresponding author. E-mail: lei@cc.ee.ntu.edu.tw .

payment schemes, or as tickets in applications such as secret voting schemes. The security of the blind signature schemes proposed in [1, 3] are based on the hardness of factorization [11] and the schemes proposed in [2, 4] is based on the hardness of computing discrete logarithm [12].

Threshold signatures [13, 14] 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.

Instead of a single signer, two blind threshold signature schemes [15] have been proposed in a distributed environment, where several signers work together to sign a blind threshold signature. The schemes proposed in [15] allows t out of n participants in a group cooperating to sign a blind threshold signature without the assistance of a single trusted authority. In these schemes, the size of a threshold signature is the same as that of an individual signature and the signature verification process is equivalent to that of an individual signature. Therefore, these schemes are optimal with respect to the threshold signature size and the verification process.

In addition to the secure voting schemes [8, 9, 10] to protect voters' privacy, the concept of blind signatures has been widely used in secure electronic payment schemes [1, 5, 6, 7]. Up to date, the on-line e-cash schemes proposed by Chaum [1, 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. These schemes involve customers, the bank and the shops and consists of the following phases: the withdrawal phase, the spending phase and the deposit phase. In real world environments, if the issue of e-coins are 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 as e-coins from t arbitrary administrators, so that, t arbitrary administrators can

represent the bank to issue e-coins. The underlying assumption is that: at least $(n-t+1)$ of the n administrators do not conspire with the others. The blind threshold signature schemes can be directly applied to these secure e-cash schemes for distributing the power of a single authority. By these schemes, secure e-cash schemes can meet the real world environments, such that, the issue of e-coins is controlled by several administrators. The blind threshold signature will work when at least t out of n administrator are honest. Since customers only need to request exact t members from n administrators, it can meet the real world environments without a single trusted administrator or with some absent/dishonest administrators.

Since blind signature schemes provide perfect unlinkability, such e-cash schemes can be misused by criminals, e.g. to safely obtain a ransom or to launder money [16]. To cope with this dilemma, the concept of fair blind signatures is introduced in [17]. In [17], three fair blind signature schemes are introduced to prevent the misuse of the unlinkability property. With the help of the judge, the signer can link a signature to the corresponding signing process. Since the fairness property is very important for preventing criminals from misusing the unlinkability property in e-cash schemes, we propose a fair blind threshold signature scheme based on the blind threshold signature scheme proposed in [15] and the registration method proposed in [17]. Our scheme allows the judge to deliver information allowing anyone of the t signers to link his view of the protocol and the message-signature pair. In our scheme, the size of a fair threshold signature is the same as that of an individual fair signature and the signature verification process is simplified by means of a group public key. The security of our schemes relies on the difficulty of computing discrete logarithm and it is computationally infeasible for signers to derive the exact correspondence between the message they actually sign and all signers' complete views of the execution of the signing process without the assistance of the judge or the requester.

The paper is organized as follows. In Section 2, we present the definition of blindness of a threshold signature scheme. In Section 3, we present an efficient fair blind threshold signature scheme. Then we examine its correctness, security and linkage recovery in Section 4. In Section 5, we make some discussions. Finally, a concluding remark is given in Section 6.

2 Preliminary

In this section, we present the definition of blindness of a threshold signature scheme. There are two methods for verifying the validity of a signature: the comparison method and the restoration (message recovery) method [18]. 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 [19, 20, 21] 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 [11, 22]. We first define the blindness of a digital signature scheme with the comparison method as follows:

Definition 1 A blind signature scheme with the comparison method is an 11-tuple $\mathcal{P} = (\mathcal{M}, \mathcal{S}, \Delta, \mathcal{K}, \Psi, \Re, \Omega, \partial, \Upsilon, \Phi, \Gamma)$, 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,*
- Ψ *is the signer of the scheme,*
- $\Re$ *is a set of requesters,*
- Ω *is a poly-time algorithm that on input a random string* $\chi \in \Delta$*, constructs a private key* $K_d \in \mathcal{K}_d$ *and its corresponding public key* $K_e \in \mathcal{K}_e$*,*
- ∂ *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* $h(\delta) \in \Delta$*, where* h *is a one-way hash function and* $\delta \in \Delta$*, constructs the blinded message* $m' = \partial(m, \lambda, K_e, h(\delta)) \in \mathcal{M}$*,*

- Υ *is a poly-time signing algorithm that on input a blinded message* $m' = \partial(m, \lambda, K_e, h(\delta)) \in \mathcal{M}$, *the private key* $K_d \in \mathcal{K}_d$ *and the randomizing factor* δ, *constructs the blind signature* $s' = \Upsilon(m', K_d, \delta) \in \mathcal{S}$ *on* m',

- Φ *is a poly-time unblinding algorithm that on input a blind signature* $s' = \Upsilon(\partial(m, \lambda, K_e, h(\delta)), K_d, \delta) \in \mathcal{S}$ *and the random blinding string* λ, *extracts the signature* $s = \Phi(s', \lambda)$ *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 signature,* Ψ *chooses a random string* $\chi \in \Delta$, *executes* $\Omega(\chi)$ *to construct a private key* $K_d \in \mathcal{K}_d$ *and its corresponding public key* $K_e \in \mathcal{K}_e$ *and then publishes his public key* K_e.

2. *In a blind signature generation, a requester* $R \in \Re$ *chooses a random string* $\lambda \in \Delta$ *and computes* $m' = \partial(m, \lambda, K_e, h(\delta))$, *where* K_e *is* Ψ*'s public key and* δ *is the randomizing factor chosen by* Ψ, *for blinding a message* m *and submits* m' *to* Ψ. Ψ *then applies the signing algorithm* Υ *to* m' *by his private key* $K_d \in \mathcal{K}_d$ *and the randomizing factor* δ *and sends the signing result* $s' = \Upsilon(m', K_d, \delta)$ *to* R. *After receiving* s', R *extracts the signature* $s = \Phi(s', \lambda)$ *on the message* m.

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

4. *In a blind signature generation, the signer' view 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.

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 2 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.* □

3 The proposed scheme

In this section, we propose a fair blind threshold signature scheme. In a typical signing process of a fair blind threshold signature scheme, there are three kinds of participants, the signers, the judge and a requester. Before the requester can obtain a signature from the signers, all the signers have to cooperate to distribute their secret shadows to other signers in advance. Then the requester acquires two pseudonyms from the judge and uses one of the pseudonyms to request a fair blind threshold signature from the signers. The proposed scheme consists of four phases: (1) the shadow distribution phase, (2) the registration phase, (3) the signature generation phase and (4) 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 registration phase, the requester requests two pseudonyms from the judge. One of the pseudonyms is used in the signature generation phase, whereas the other one is part of the signature. Thus, the judge, who knows the two corresponding pseudonyms, can link the message-signature pair with the corresponding signer's view. In the signature generation phase, a requester requests a blind threshold signature from the signers by sending the pseudonym to the signers and the signers cooperate to issue the fair blind threshold signature to the requester. In the signature verification phase, anyone can use the group public key to verify if a fair 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 fair blind threshold signature scheme, so that at least $(n - t + 1)$ signers are

honest, m be the blind message to be signed, h be a secure one-way hashing function [23], p and q be two large strong prime numbers such that q divides $(p-1)$, and ρ be a generator of Z_p^* (*i.e.*, $\gcd(\rho, p) = 1$, $\rho \neq 1$). Let $x \equiv_p y$ denote $x = y \bmod p$. Let $g \equiv_p \rho^{(p-1)/q}$ and " $\cdot$ " denote the ordinal string concatenation. Let d_i be the secret key chosen by U_i and d_J be the secret key chosen by the judge. In a distributed environment, U_i and the judge can publish their corresponding public keys e_i and e_J. Anyone can get e_i and e_J via some authentication service (e.g. the X.509 directory authentication service [23]). Using a secure public key signature scheme [11, 12], U_i and the judge can produce signatures of messages by their own secret keys d_i and d_J. Anyone can verify these signatures by the corresponding public keys e_i and e_J. Let $Cert_{U_i}(m)$ be the signature on the message m produced by U_i and $Cert_J(m)$ be the signature on the message m produced by the judge.

3.1 The shadow distribution phase

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

1. U_i chooses a secret key $z_i \in Z_q$ and a secret polynomial $f_i(x) = \sum_{k=0}^{t-1} a_{i,k} x^k$ such that $a_{i,0} = z_i$, computes $\Psi_{i,k} \equiv_p g^{a_{i,k}}$ and the signatures $Cert_{U_i}(h(\Psi_{i,k}))$ on $\Psi_{i,k}$ for $0 \leq k \leq t-1$ and sends $((\Psi_{i,k}, Cert_{U_i}(h(\Psi_{i,k})), 0 \leq k \leq t-1)$ to U_j, $1 \leq j \leq n$, $j \neq i$.

2. Upon receiving $((\Psi_{j,k}, Cert_{U_j}(h(\Psi_{j,k})), 1 \leq j \leq n, j \neq i, 0 \leq k \leq t-1)$ from all other signers, U_i verifies if all $Cert_{U_j}(h(\Psi_{j,k}))$ are valid. If valid, he sends $\delta_{i,j} \equiv_q f_i(x_j)$, where x_j is a unique public number for U_j, and a signature $Cert_{U_i}(h(\delta_{i,j}))$ on $\delta_{i,j}$ secretly to every U_j, $1 \leq j \leq n$, $j \neq i$. Otherwise, he publishes the invalid signatures and stops.

3. When U_i receives all $\delta_{j,i}$, $Cert_{U_j}(h(\delta_{j,i}))$, $1 \leq j \leq n$, $j \neq i$, from other signers, he verifies if the share $\delta_{j,i}$ received from U_j is consistent with the certified values $\Psi_{j,l}$, $0 \leq l \leq t-1$, by checking whether $g^{\delta_{j,i}} \equiv_p \prod_{l=0}^{t-1} (\Psi_{j,l})^{x_i{}^l}$. If it fails, U_i broadcasts that an error has been found, publishes $\delta_{j,i}$, $Cert_{U_j}(h(\delta_{j,i}))$ 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 $y \equiv_p \prod_{l=1}^{n} y_l \equiv_p \prod_{l=1}^{n} \Psi_{l,0}$ and the signature $Cert_{U_i}(h(\Phi_{j,i}))$ on $\Phi_{j,i} \equiv_p g^{\delta_{j,i}}$, $1 \leq j \leq n$. He then sends $(Cert_{U_i}(h(y)), (\Phi_{j,i}, Cert_{U_i}(h(\Phi_{j,i})), 1 \leq j \leq n))$ to all other signers.

4. Upon receiving all $((Cert_{U_j}(h(y)), 1 \leq j \leq n, j \neq i), (\Phi_{l,j}, Cert_{U_j}(h(\Phi_{l,j})), 1 \leq l \leq n, 1 \leq j \leq n, j \neq i))$, U_i verifies if all $((Cert_{U_j}(h(y)), 1 \leq j \leq n, j \neq i), (Cert_{U_j}(h(\Phi_{l,j})), 1 \leq l \leq n, 1 \leq j \leq n, j \neq i))$ are valid. If valid, the shadow keys corresponding to the group secret key $z \equiv_q \sum_{j=1}^{n} z_j$ have been securely and correctly distributed. The group public key $y \equiv_p \prod_{j=1}^{n} y_j \equiv_p g^{\sum_{j=1}^{n} z_j}$, all signers' public keys y_j, $1 \leq j \leq n$, and all public shadows $\Phi_{l,j} \equiv_p g^{\delta_{l,j}}$, $1 \leq l, j \leq n$, can then be published by each signer. Otherwise, U_i publishes the invalid signatures and stops.

3.2 The registration phase

Before a requester requests a fair blind threshold signature from the signers, he must acquire two pseudonyms from the judge by performing the following steps.

1. The requester sends a request for two pseudonyms to the judge.

2. The judge randomly chooses η and $\gamma \in Z_q$, computes $\Omega_0 \equiv_p g^{\eta}$ and $\Omega_1 \equiv_p \Omega_0^{\gamma}$, stores $(\gamma, \Omega_0, \Omega_1)$ and then sends the random numbers, the pseudonyms and their signatures $(\eta, \gamma, \Omega_0, \Omega_1, Cert_J(h(\Omega_0)), Cert_J(h(\Omega_1)), Cert_J(h(\eta \cdot \gamma \cdot \Omega_0 \cdot \Omega_1)))$ back to the requester.

3. Upon receiving the random numbers, the pseudonyms and their signatures $(\eta, \gamma, \Omega_0, \Omega_1, Cert_J(h(\Omega_0)), Cert_J(h(\Omega_1)), Cert_J\ (h(\eta \cdot \gamma \cdot \Omega_0 \cdot \Omega_1)))$, the requester verifies if $\Omega_0 \equiv_p g^{\eta}$, $\Omega_1 \equiv_p \Omega_0^{\gamma}$ and the signatures of the pseudonyms are valid. If not, he has to ask the judge to retransmit them.

3.3 The signature generation phase

Without loss of generality, we assume that t out of the n signers are U_i, $1 \leq i \leq t$. When a requester requests a fair blind threshold signature, he and the t signers perform the following steps during the signature generation phase.

1. The requester sends $\Omega_0, Cert_J(h(\Omega_0))$ to all $U_i, 1 \leq i \leq t$.

2. Upon receiving $\Omega_0, Cert_J(h(\Omega_0))$, each U_i verifies if $Cert_J(h(\Omega_0))$ is valid by the judge's public key e_J. If valid, each U_i randomly chooses a number $k_i \in Z_q$, computes $\widehat{r_i} \equiv_p g^{k_i}$, $\Gamma_i \equiv_p \Omega_0^{k_i}$, $u_i \equiv_p \Omega_0^{z_i + \sum_{j=t+1}^{n} f_j(x_i)(\prod_{k=1,k\neq i}^{t}(\frac{-x_k}{x_i - x_k}))}$ and sends $\widehat{r_i}$, Γ_i and u_i to the requester. Otherwise, he rejects it and stops.

3. After receiving all $\widehat{r_i}$, Γ_i and $u_i, 1 \leq i \leq t$, the requester checks if

$$u_i \equiv_p (\Psi_{i,0}(\prod_{j=t+1}^{n} \Phi_{j,i})^{\prod_{k=1,k\neq i}^{t}(\frac{-x_k}{x_i - x_k})})^{\eta}, 1 \leq i \leq t. \quad (1)$$

   If $u_i, 1 \leq i \leq t$, is not valid he has to ask the corresponding signer to send it again. Otherwise, he does the following.

   (a) Choose two random numbers $\alpha \in Z_q$ and $\beta \in Z_q^*$, compute $u \equiv_p (\prod_{i=1}^{t} u_i)^{\gamma}$, $\Gamma \equiv_p \prod_{i=1}^{t} \Gamma_i$, $r_i \equiv_p g^{\alpha} \widehat{r_i}^{\beta}, v_2 \equiv_p (\Omega_1)^{(t\alpha)} \Gamma^{\gamma\beta}$, $v_1 \equiv_p h(m \cdot \Omega_1 \cdot v_2 \cdot u) \prod_{i=1}^{t} r_i$ and $\widehat{m} \equiv_q \beta^{-1} v_1$.

   (b) Check if $\widehat{m} \neq 0$. If yes, send $\widehat{m}$ to all $U_i, 1 \leq i \leq t$. Otherwise, go back to step (a).

4. Upon receiving $\widehat{m}$, each U_i computes

$$\widehat{s_i} \equiv_q \widehat{m}(z_i + \sum_{j=t+1}^{n} f_j(x_i)(\prod_{k=1,k\neq i}^{t}(\frac{-x_k}{x_i - x_k}))) + k_i \quad (2)$$

   and sends $\widehat{s_i}$ back to the requester.

5. After receiving all $\widehat{s_i}$, $1 \leq i \leq t$, the requester computes $s_i \equiv_q \widehat{s_i}\beta + \alpha, 1 \leq i \leq t$, and checks if

$$g^{-s_i} y_i^{v_1} r_i \equiv_p (\prod_{j=t+1}^{n} (\Phi_{j,i}))^{(\prod_{k=1,k\neq i}^{t}(\frac{-x_k}{x_i - x_k}))(-v_1)}, 1 \leq i \leq t. \quad (3)$$

If $\widehat{s_i}, 1 \leq i \leq t$, is not valid, he has to ask the corresponding signer to send it again. Otherwise, he computes $s \equiv_q \sum_{i=1}^{t} s_i$. The fair threshold signature of m is $(\Omega_1, Cert_J(h(\Omega_1)), v_1, v_2, s, u)$.

3.4 The signature verification phase

To verify the fair threshold signature $(\Omega_1, Cert_J(h(\Omega_1)), v_1, v_2, s, u)$ for the message m, one simply checks if $\Omega_1^s \equiv_p v_2 u^{v_1}$ and $g^{-s} y^{v_1} v_1 \equiv_p h(m \cdot \Omega_1 \cdot v_2 \cdot u)$.

4 Analysis

We examine the correctness and security of our scheme in this section. We also show how to link a given signature to its corresponding signing process under the assistance of the judge.

4.1 Correctness

To prevent a signer from sending an invalid partial signature to the requester, the partial signature must be checked in step 5 of the signature generation phase. The following lemma ensures the correctness of partial signatures.

Lemma 1. *The partial signature (r_i, s_i, u_i) is valid if U_i is honest.*

Proof. By our scheme, we have

$$
\begin{aligned}
& g^{-s_i} y_i^{v_1} r_i \\
\equiv_p\ & g^{-(\widehat{s}_i\beta+\alpha)} g^{z_i v_1} g^{\alpha} \widehat{r}_i^{\beta} \\
\equiv_p\ & g^{-(\widehat{m}(z_i+\sum_{j=t+1}^{n} f_j(x_i)(\prod_{k=1,k\neq i}^{t}(\frac{-x_k}{x_i-x_k})))+k_i)\beta} g^{z_i v_1} g^{k_i\beta} \\
\equiv_p\ & g^{-\widehat{m}(z_i+\sum_{j=t+1}^{n} f_j(x_i)(\prod_{k=1,k\neq i}^{t}(\frac{-x_k}{x_i-x_k})))\beta} g^{z_i v_1} \\
\equiv_p\ & g^{-\widehat{m}z_i\beta-\widehat{m}\sum_{j=t+1}^{n} f_j(x_i)(\prod_{k=1,k\neq i}^{t}(\frac{-x_k}{x_i-x_k}))\beta} g^{z_i v_1} \\
\equiv_p\ & g^{\sum_{j=t+1}^{n} f_j(x_i)(\prod_{k=1,k\neq i}^{t}(\frac{-x_k}{x_i-x_k}))(-\widehat{m}\beta)} \\
\equiv_p\ & (\prod_{j=t+1}^{n}(\Phi_{j,i}))^{(\prod_{k=1,k\neq i}^{t}(\frac{-x_k}{x_i-x_k}))(-v_1)}
\end{aligned}
$$

and

$$
\begin{aligned}
& u_i \\
\equiv_p\ & \Omega_0^{z_i+\sum_{j=t+1}^{n} f_j(x_i)(\prod_{k=1,k\neq i}^{t}(\frac{-x_k}{x_i-x_k}))} \\
\equiv_p\ & (g^{\eta})^{z_i+\sum_{j=t+1}^{n} f_j(x_i)(\prod_{k=1,k\neq i}^{t}(\frac{-x_k}{x_i-x_k}))} \\
\equiv_p\ & g^{(z_i+\sum_{j=t+1}^{n} f_j(x_i)(\prod_{k=1,k\neq i}^{t}(\frac{-x_k}{x_i-x_k})))\eta} \\
\equiv_p\ & (g^{z_i+\sum_{j=t+1}^{n} f_j(x_i)(\prod_{k=1,k\neq i}^{t}(\frac{-x_k}{x_i-x_k}))})^{\eta} \\
\equiv_p\ & (\Psi_{i,0}(\prod_{j=t+1}^{n}\Phi_{j,i})^{\prod_{k=1,k\neq i}^{t}(\frac{-x_k}{x_i-x_k})})^{\eta}.
\end{aligned}
$$

□

After the signature generation phase, the blind signature can be verified by the group public key in the signature verification phase. Lemma 2 ensures the correctness of the scheme.

Lemma 2. *The 6-tuple* $(\Omega_1, Cert_J(h(\Omega_1)),\ v_1, v_2, s, u)$ *is a valid fair blind threshold signature on the message* m.

Proof. The validity of the signature $(\Omega_1, Cert_J(h(\Omega_1)),\ v_1, v_2, s, u)$ on the message m can easily be established as follows.

$$
\begin{aligned}
& g^{-s} y^{v_1} v_1 \\
\equiv_p\ & g^{-(\sum_{i=1}^{t}(\widehat{s}_i\beta+\alpha))} g^{\sum_{i=1}^{n} z_i v_1} h(m\cdot\Omega_1\cdot v_2\cdot u)(\prod_{i=1}^{t} r_i) \\
\equiv_p\ & h(m\cdot\Omega_1\cdot v_2\cdot u) g^{-(\widehat{m}(\sum_{i=1}^{t} z_i+\sum_{i=1}^{t}(\sum_{j=t+1}^{n} f_j(x_i)(\prod_{k=1,k\neq i}^{t}(\frac{-x_k}{x_i-x_k}))))+\sum_{i=1}^{t} k_i)\beta-t\alpha} g^{\sum_{i=1}^{n} z_i v_1} \\
& (\prod_{i=1}^{t} g^{\alpha}\widehat{r}_i{}^{\beta}) \\
\equiv_p\ & h(m\cdot\Omega_1\cdot v_2\cdot u) g^{-(\widehat{m}(\sum_{i=1}^{t} z_i+\sum_{j=t+1}^{n}(\sum_{i=1}^{t} f_j(x_i)(\prod_{k=1,k\neq i}^{t}(\frac{-x_k}{x_i-x_k}))))+\sum_{i=1}^{t} k_i)\beta} g^{\sum_{i=1}^{n} z_i v_1} \\
& (\prod_{i=1}^{t} g^{k_i\beta}) \\
\equiv_p\ & h(m\cdot\Omega_1\cdot v_2\cdot u) g^{-(\widehat{m}(\sum_{i=1}^{t} z_i+\sum_{i=t+1}^{n} z_i))\beta} g^{\sum_{i=1}^{n} z_i v_1} \\
\equiv_p\ & h(m\cdot\Omega_1\cdot v_2\cdot u) g^{-\widehat{m}\sum_{i=1}^{n} z_i\beta} g^{\sum_{i=1}^{n} z_i v_1} \\
\equiv_p\ & h(m\cdot\Omega_1\cdot v_2\cdot u) g^{-v_1\sum_{i=1}^{n} z_i} g^{\sum_{i=1}^{n} z_i v_1} \\
\equiv_p\ & h(m\cdot\Omega_1\cdot v_2\cdot u)
\end{aligned}
$$

and

$$
\begin{aligned}
& \Omega_1^s \\
\equiv_p\ & \Omega_0^{\gamma(\sum_{i=1}^{t} s_i)} \\
\equiv_p\ & \Omega_0^{\gamma(\sum_{i=1}^{t}(\widehat{s}_i\beta+\alpha))} \\
\equiv_p\ & \Omega_0^{\gamma t\alpha+\gamma\beta\sum_{i=1}^{t}\widehat{s}_i} \\
\equiv_p\ & \Omega_0^{\gamma t\alpha+\gamma\beta\sum_{i=1}^{t}(\widehat{m}(z_i+\sum_{j=t+1}^{n} f_j(x_i)(\prod_{k=1,k\neq i}^{t}(\frac{-x_k}{x_i-x_k})))+k_i)} \\
\equiv_p\ & \Omega_0^{\gamma t\alpha+\gamma\beta\sum_{i=1}^{t} k_i+\gamma\beta\widehat{m}\sum_{i=1}^{t}(z_i+\sum_{j=t+1}^{n} f_j(x_i)(\prod_{k=1,k\neq i}^{t}(\frac{-x_k}{x_i-x_k})))} \\
\equiv_p\ & \Omega_0^{\gamma t\alpha+\gamma\beta\sum_{i=1}^{t} k_i+\gamma v_1\sum_{i=1}^{n} z_i} \\
\equiv_p\ & (\Omega_1)^{(t\alpha)}(\Omega_1)^{(\beta\sum_{i=1}^{t} k_i)}(\Omega_1)^{(v_1\sum_{i=1}^{n} z_i)} \\
\equiv_p\ & (\Omega_1)^{(t\alpha)}(\Omega_1)^{(\beta\sum_{i=1}^{t} k_i)}(u)^{v_1} \\
\equiv_p\ & (\Omega_1)^{(t\alpha)}\Gamma^{\gamma\beta}u^{v_1} \\
\equiv_p\ & v_2 u^{v_1}.
\end{aligned}
$$

□

4.2 Security analysis

Let ν denote the signers' complete views of an execution in the signature generation phase and let $(m, (\Omega_1, Cert_J\ (h(\Omega_1)), v_1, v_2, s, u)$ denote the message-signature pair generated in that execution. Theorem 3 ensures the blindness of our proposed scheme.

Theorem 3. *The threshold signature scheme proposed in Section 2 is blind.*

Proof. For proving the blindness of the scheme, we show that given any view ν and any valid message-signature pair $(m, (\Omega_1, Cert_J(h(\Omega_1)), v_1, v_2, s, u))$, there exists a unique triple of blinding factors α, β and γ. Since the requester chooses the blinding factors α and β randomly and the judge also chooses the blinding factor γ randomly, the blindness of the signature scheme follows.

Without loss of generality, assume that the signature $(\Omega_1, Cert_J(h(\Omega_1)),\ v_1, v_2, s, u)$ for the message m has been generated by t signers U_i, $1 \leq i \leq t$, with the view consisting of $\Omega_0, k_i,\ \widehat{r}_i \equiv_p g^{k_i}, \Gamma_i \equiv_p \Omega_0^{k_i},\ \widehat{s}_i \equiv_q \widehat{m}(z_i + \sum_{j=t+1}^{n} f_j(x_i)(\prod_{k=1,k\neq i}^{t}(\frac{-x_k}{x_i-x_k}\ ))) + k_i,\ u_i \equiv_p \Omega_0^{z_i+\sum_{j=t+1}^{n} f_j(x_i)(\prod_{k=1,k\neq i}^{t}(\frac{-x_k}{x_i-x_k}))}$, $1 \leq i \leq t$ and $\widehat{m}$, then the following equations must hold for α and β.

$$v_1 \equiv_p m\prod_{i=1}^{t} r_i \equiv_p m\prod_{i=1}^{t} g^{\alpha}\widehat{r}_i^{\beta} \tag{4}$$

$$\widehat{m} \equiv_q v_1\beta^{-1} \tag{5}$$

$$s \equiv_q \sum_{i=1}^{t} s_i \equiv_q \sum_{i=1}^{t}(\widehat{s}_i\beta + \alpha) \tag{6}$$

Note that if $t < q$, then $\gcd(t, q) = 1$. Since $\widehat{m} \in Z_q$ and $\widehat{m} \neq 0$, by equations (5) and (6), the unique solution for α and β is:

$$\beta \equiv_q \widehat{m}^{-1}v_1 \tag{7}$$

$$\alpha \equiv_q (s - \sum_{i=1}^{t} \widehat{s}_i\beta)t^{-1} \tag{8}$$

In the following, we show that the solutions of α and β in equations (7) and (8) also satisfies equation (4).

$h(m \cdot \Omega_1 \cdot v_2 \cdot u) \prod_{i=1}^{t} g^{\alpha \widehat{r_i}^{\beta}}$

$\equiv_p g^{-s} y^{v_1} v_1 g^{t\alpha} \prod_{i=1}^{t} g^{k_i \beta}$

$\equiv_p v_1 g^{-\sum_{i=1}^{t} (\widehat{s_i}\beta + \alpha)} g^{v_1 \sum_{i=1}^{n} z_i} g^{t\alpha} g^{\beta \sum_{i=1}^{t} k_i}$

$\equiv_p v_1 g^{-((\widehat{m}(\sum_{i=1}^{t} z_i + \sum_{i=1}^{t}(\sum_{j=t+1}^{n} f_j(x_i)(\prod_{k=1,k\neq i}^{t}(\frac{-x_k}{x_i - x_k})))) + \sum_{i=1}^{t} k_i)\beta + t\alpha)} g^{v_1 \sum_{i=1}^{n} z_i} g^{t\alpha} g^{\beta \sum_{i=1}^{t} k_i}$

$\equiv_p v_1 g^{-(\widehat{m}(\sum_{i=1}^{t} z_i + \sum_{i=t+1}^{n} z_i) + \sum_{i=1}^{t} k_i)\beta} g^{v_1 \sum_{i=1}^{n} z_i} g^{\beta \sum_{i=1}^{t} k_i}$

$\equiv_p v_1 g^{-((\widehat{m} \sum_{i=1}^{n} z_i + \sum_{i=1}^{t} k_i)\beta)} g^{v_1 \sum_{i=1}^{n} z_i} g^{\beta \sum_{i=1}^{t} k_i}$

$\equiv_p v_1 g^{-\sum_{i=1}^{n} \widehat{m} z_i \beta} g^{v_1 \sum_{i=1}^{n} z_i}$

$\equiv_p v_1.$

In additional to the equations (4), (5) and (6), the following equations must hold for γ.

$$\Omega_1 \equiv_p \Omega_0^{\gamma} \tag{9}$$

$$u \equiv_p (\prod_{i=1}^{t} u_i)^{\gamma} \tag{10}$$

$$v_2 \equiv_p (\Omega_0^{\gamma})^{(t\alpha)} \Gamma^{\gamma\beta} \tag{11}$$

Since $g \equiv_p \rho^{(p-1)/q}$ and ρ is a generator of Z_p^*, g generates a cyclic subgroup S_g of Z_p^* with $|S_g| = q$ and $\Omega_0, \Omega_1 \in S_g$, we can only find a unique solution for γ satisfying equation (9). This unique solution γ also satisfies equation (10) and (11). □

Given the secret information of a group of $\sigma < t$ members, Lemma 4 ensures that the threshold cryptosystem constructed in the shadow distribution phase will not disclose any extra information about the group secret key $\sum_{i=1}^{n} z_i$.

Lemma 4. *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 $\{\delta_{j,i} | 1 \leq j \leq n,\ i \in G\}$. For any fixed j, $1 \leq j \leq n$, it takes polynomial time on $|p|$ to generate a random set $\{g^{\widehat{a_{j,k}}} | 1 \leq k \leq t-1\}$ satisfying $g^{\delta_{j,i}} \equiv_p \prod_{k=0}^{t-1} (g^{\widehat{a_{j,k}}})^{x_i{}^k}$ 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

$$g^{\delta_{j,i}} \equiv_p \prod_{l=0}^{t-1}(g^{a_{j,l}})^{{x_i}^l} \equiv_p g^{\sum_{l=0}^{t-1} a_{j,l} * {x_i}^l}. \tag{12}$$

Since $g \equiv_p \rho^{(p-1)/q}$ and ρ is a generator of Z_p^*, g generates a cyclic subgroup S_g of Z_p^* with $|S_g| = q$. From (12), we have

$$\delta_{j,i} \equiv_q \sum_{l=0}^{t-1} a_{j,l} * {x_i}^l \tag{13}$$

From (13), 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:

$$\delta_{j,i} \equiv_q \sum_{k=0}^{t-1} \widehat{a_{j,k}} * {x_i}^k. \tag{14}$$

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

$$\delta_{j,i} \equiv_q \sum_{k=0}^{t-1} \widehat{a_{j,k}} * {x_i}^k (i \in G). \tag{15}$$

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 (15) and get a random solution $\widehat{a_{j,k}}$, $1 \leq k \leq t-1$, by assigning random values to all free variables. From (15), it is clear that $g^{\delta_{j,i}} \equiv_p g^{\sum_{k=0}^{t-1} \widehat{a_{j,k}} * {x_i}^k} \equiv_p \prod_{k=0}^{t-1}(g^{\widehat{a_{j,k}}})^{{x_i}^k}$. □

In our fair blind threshold signature scheme, the partial signature (s_i, r_i, u_i) must satisfy the equation $g^{-s_i} y_i^{v_1} r_i \equiv_p g^{-s_i} g^{z_i v_1} r_i \equiv_p (\prod_{j=t+1}^{n}(\Phi_{j,i}))^{(\prod_{k=1,k\neq i}^{t}(\frac{-x_k}{x_i - x_k}))(-v_1)}$ and $g^{u_i} \equiv_p (\Psi_{i,0}(\prod_{j=t+1}^{n}\Phi_{j,i})^{\prod_{k=1,k\neq i}^{t}(\frac{-x_k}{x_i-x_k})})^{\Omega_0}$. Since v_1, $\Phi_{j,i}$, x_k, r_i, y_i and s_i are all public, an attacker has to solve the discrete logarithm problem in order to get the secret value z_i.

With the information of all partial signatures and the corresponding threshold signature, an attacker is not capable of deriving the secret keys since it has to solve the equation $v_1 g^{-s} y^{v_1} \equiv_p h(m \cdot \Omega_1 \cdot v_2 \cdot u)(\prod_{i=1}^{n} r_i) g^{-(\sum_{i=1}^{n} s_i)}(\prod_{i=1}^{n} g^{z_i})^{v_1}$. To solve this equation, one has to solve the discrete logarithm problem.

Since γ, α and β are kept secret by the requester and all signatures are equally likely from the signer's point of view, it is computationally infeasible for the signer to derive the link between the view consisting of Ω_0, k_i, $\widehat{r_i} \equiv_p g^{k_i}, \Gamma_i \equiv_p \Omega_0^{k_i}$, $\widehat{s_i} \equiv_q \widehat{m}(z_i + \sum_{j=t+1}^{n} f_j(x_i)(\prod_{k=1,k\neq i}^{t}(\frac{-x_k}{x_i-x_k}))) + k_i$, $u_i \equiv_p \Omega_0^{z_i + \sum_{j=t+1}^{n} f_j(x_i)(\prod_{k=1,k\neq i}^{t}(\frac{-x_k}{x_i-x_k}))}, 1 \leq i \leq t$,

$\widehat{m}$ and the signature $(\Omega_1 \equiv_p \Omega_0^{\gamma}, Cert_J(h(\Omega_1)),\ v_1, v_2, s, u)$ for the message m submitted by a requester for verification later.

4.3 Linkage recovery

Since blind threshold signature schemes without the fairness property provide perfect unlinkability, such e-cash schemes can be misused by criminals, e.g. to safely obtain a ransom or to launder money. For example, a criminal can safely obtain a ransom by joining a blind threshold signature scheme where the request is via an untraceable mail (e.g. an ordinary mail or an untraceable e-mail [24, 25]) and the signers put the blind threshold signature on a public board. Then the criminal can easily obtain the blind threshold signature from the public board and derive the corresponding e-coins. To cope with this dilemma, in our proposed scheme, anyone of the t signers can first send all pseudonyms $(\Omega_0, Cert_J(h(\Omega_0)))_s$ requested by the criminal to the judge and then the judge sends all the corresponding pseudonyms $(\gamma, \Omega_0, \Omega_1, Cert_J(h(\Omega_0)), Cert_J(h(\Omega_1)))_s$ back to the signer. The signer can verify validity of the corresponding pseudonyms by checking if $\Omega_0^{\gamma} \equiv_p \Omega_1$ and both $Cert_J(h(\Omega_0))$ and $Cert_J(h(\Omega_1))$ are valid. When the criminal withdraws these e-coins from the signer, the signer can easily identify the criminal by linking the message-signature pair $(m, (\Omega_1, Cert_J\ (h(\Omega_1)),\ v_1, v_2, s, u))$ with the corresponding signer's view $\Omega_0, k_i,\ \widehat{r_i} \equiv_p g^{k_i}, \Gamma_i \equiv_p \Omega_0^{k_i},\ \widehat{s_i} \equiv_q \widehat{m}(z_i + \sum_{j=t+1}^{n} f_j(x_i)\ (\prod_{k=1,k\neq i}^{t}(\frac{-x_k}{x_i - x_k}$ $))) + k_i,\ u_i \equiv_p \Omega_0^{z_i + \sum_{j=t+1}^{n} f_j(x_i)(\prod_{k=1,k\neq i}^{t}(\frac{-x_k}{x_i - x_k}))}$ and $\widehat{m}$. If the judge is honest, all crimes by misusing the unlinkability property of blind threshold signatures will be prevented and the anonymity of honest customers will also be preserved.

5 Discussions

5.1 Performance Considerations

In this subsection we give an analysis of the computational effort required to compute fair blind threshold signatures in our scheme. Let Scheme 1 denote the fair blind threshold signature scheme in Section 3.3.1 and Scheme 1* denote the corresponding underlying blind signature scheme. Table 1 illustrates the comparison of the fair blind threshold signature scheme and the underlying fair blind signature scheme. Comparative to the underlying

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

	The requester				The Signer or U_i			
	EXP	INV	MUL	ADD	EXP	INV	MUL	ADD
Scheme 1	5	1	$3t+6$	t	3	0	$n-1$	$n-t+1$
Scheme 1*	5	1	5	1	3	0	1	1

where
EXP = the number of modulo exponentiations,
INV = the number of modulo inversions (divisions),
MUL = the number of modulo multiplications,
ADD = the number of modulo additions.

blind signature scheme, the extra cost for signing a blind threshold signature is to compute $z_i+\sum_{j=t+1}^{n} f_j(x_i)(\prod_{k=1,k\neq i}^{t}(\frac{-x_k}{x_i-x_k}))$ in Step 3 which contains $n-2$ modular multiplications and $n-t$ additions. For reducing the computational cost needed by the requester, the partial signature verification in Step 3 and Step 5 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 compute 5 modular exponentiations and 1 modular inverse in step 2 of the signature generation phase which is the same as the underlying fair blind signature scheme. Since the blind threshold verification functions of our schemes all are the same as those of the underlying blind signature schemes, the verification cost of our blind threshold signature is the same as that of the underlying blind signature. Comparative to the underlying fair blind signature schemes, the extra cost for requesting a fair blind threshold signature in our scheme is to compute $\prod_{i=1}^{t}\Gamma_i, t\alpha, \prod_{i=1}^{t}\widehat{r}_i$, $\gamma\beta, \prod_{i=1}^{t}u_i$ and $\sum_{i=1}^{t}\widehat{s}_i$ in the equation $s \equiv_q \sum_{i=1}^{t}s_i \equiv_q \sum_{i=1}^{t}(\widehat{s}_i\beta+\alpha) \equiv_q t\alpha+\beta\sum_{i=1}^{t}\widehat{s}_i$ which contains $3t+1$ modular multiplications and $t-1$ modular additions.

5.2 Message Recovery

It is clear that our protocol can not provide the message recovery capability. Since our proposed blind threshold signature scheme is based on the blind signature scheme in [2] with message recovery. We can slightly modify our proposed scheme, such that, the modified scheme provides the message recovery capability. The verification process of the modified scheme can be the as follows:

1. Checks if $\Omega_1^s \equiv_p v_2 u^{v_1}$.

2. Computes $m \equiv_p g^{-s} y^{v_1} v_1$ and checks if m has some proper redundancy information.

But the modified scheme fails to achieve the fairness property as will be explained below.

Let (Ω_0, Ω_1) and (Ω_0', Ω_1') be two pair of pseudonyms and let $(\Omega_1, Cert_J(h(\Omega_1)), v_1, v_2, s, u)$ be the signature generated by the pseudonym (Ω_0, Ω_1) on the message m. The signature is valid if $m \equiv_p g^{-s} y^{v_1} v_1$, m has some proper redundancy information and $\Omega_1^s \equiv_p v_2 u^{v_1}$ holds. From the signature, we can compute the signature $(\Omega_1', Cert_J(h(\Omega_1')), v_1, v_2', s, u)$ by computing $v_2' \equiv_p (\Omega_1')^s u^{-v_1}$. This is a valid signature on m since $m \equiv_p g^{-s} y^{v_1} v_1$ and $\Omega_1^s \equiv_p v_2 u^{v_1}$ holds. However, if the requester did not use Ω_1', this signature can no longer be linked to any run of the signature-generation protocol. Therefore, the modified scheme does not enjoy the fairness property. It is still an open problem if there exists a fair blind signature scheme with message recovery using the registration method.

6 Conclusion

We have proposed an efficient fair blind threshold signature scheme based on discrete logarithm. In our scheme, the size of a fair threshold signature is the same as that of an individual fair signature and the and the signature verification process is simplified by means of a group public key. The security of our schemes relies on the hardness of computing discrete logarithm and it is computationally infeasible for the signers to derive the exact correspondence between the message they actually sign and all signers' complete views of the execution of the signing process without the assistance of the judge or the requester. 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] **Chaum, D** Blind signatures for untraceable payments, *Proc. of Crypt'82*, Plenum, NY, (1983) 99-203.

[2] **Camenisch, J L, Pivereau J M and Stadler, M A** Blind signatures based on the discrete logarithm problem, *Proc. of EuroCrypt'94*, LNCS 950, Springer-Verlag (1995) 428-432.

[3] **Fan, C and Lei, C** User Efficient Blind Signatures, *IEE Electronics Letters*, 34(6) (1998) 544-546.

[4] **Horster, P, Michels, M and Petersen, H** Meta-message recovery and meta-blind signature schemes based on the discrete logarithm problem and their applications, *Proc. of AsiaCrypt'94*, LNCS 917, Springer-Verlag (1994) 224-237.

[5] **Chaum, D** Privacy protected payments: unconditional payer and/or payee untraceability, *In Smartcard 2000*, North Holland (1988).

[6] **Ferguson, N** Single term off-line coins, *Proc. of EuroCrypt'93*, LNCS 765, Springer-Verlag (1993) 318-328.

[7] **Okamoto, T and Ohta, K** Universal Electronic cash, *Proc. of Crypt'91*, LNCS 576, Springer-Verlag (1992) 324-337.

[8] **Fujioka, A, Okamoto, T and Ohta, K** A practical secret voting scheme for large scale elections, *Proc. of AusCrypt'92*, LNCS 718, Springer-Verlag (1992) 244-251.

[9] **Juang, W and Lei, C** A collision free secret ballot protocol for computerized general elections, *Computers & Security*, 15(4) (1996) 339-348.

[10] **Juang, W and Lei, C** A secure and practical electronic voting scheme for real world environments, *IEICE Trans. on Fundamentals*, E80-A(1) (January 1997) 64-71.

[11] **Rivest, R L, Shamir, A and Adelman, L** A method for obtaining digital signatures and public key cryptosystem, *Comm. of ACM*, 21(2) (1978) 120-126.

[12] **ElGamal, T** A public key cryptosystem and a signature scheme based on discrete logarithm, *IEEE Trans. on Information Theory*, IT-31(4) (1985) 469-472.

[13] **Gennaro, R, Jarecki, S, Krawczyk, H and Rabin, T** Robust threshold DSS signatures, Proc. of EuroCrypt '96, LNCS 1070, Springer Verlag (1996) 354-371.

[14] **Harn, L** Group-oriented (t, n) threshold digital signature scheme and digital multisignature, *IEE Proc. Compu. Digit. Tech.*, 141(5) (1994) 307- 313.

[15] **Juang, W and Lei, C** Blind threshold signatures based on discrete logarithm, *Proc. of Second Asian Computing Science Conference on Programming, Concurrency and Parallelism, Networking and Security*, LNCS 1179, Springer-Verlag (1996) 172 -181.

[16] **Solms, S and Naccache, D** On blind signatures and perfect crime, *Computer & Security*, 11 (1992) 581-583.

[17] **Stadler, M, Piveteau, J and Camenisch, J** Fair blind signatures, *Proc. of EuroCrypt'95,* LNCS 921, Springer-Verlag (1995) 209-219.

[18] **Okamoto, T** A digital multisignature scheme using bijective public-key cryptosystems, *ACM Trans. Computer Systems*, 6(8) (1988) 432-441.

[19] NIST FIPS PUB 180, Secure hash standard, National Institute of Standards and Technology, U. S. Department of Commerce, DRAFT (1993).

[20] **Pohlig, S and Hellman, M E** An improved algorithm for computing logarithms over GF(p) and its cryptographic significance, *IEEE Trans. on Information Theory*, IT-24 (1978) 106-110.

[21] **Rivest, R L** The MD5 message-digest algorithm, RFC 1321, Internet Activities Board, Internet Privacy Task Force (1992).

[22] **Nyberg K and Rueppel, R A** Message recovery for signature schemes based on the discrete logarithm problem, *Advances in Cryptology: Proc. of EuroCrypt'94*, LNCS 950, Springer-Verlag (1995) 182-193.

[23] **Stallings, W** *Network and internetwork security*, Prentice Hall International (1995).

[24] **Chaum, D** Untraceable electronic mail, return addresses, and digital pseudonyms, *Comm. of ACM*, 24(2) (1981) 84-88.

[25] **Juang, W, Lei, C and Fan, C** Anonymous Channel and Authentication in Wireless Communications, *to appear in Computer Communications.*