ELSEVIER

Computer Communications 22 (1999) 73–86

computer communications

Partially blind threshold signatures based on discrete logarithm

W.-S. Juang, C.-L. Lei*

Department of Electrical Engineering, Rm. 343, National Taiwan University, Taipei, Taiwan, ROC

Received 18 May 1998; accepted 22 June 1998

Abstract

In this paper, we propose a group-oriented partially 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 partially blind threshold signatures, which can be used in anonymous digital e-cash systems or secure voting schemes. By our proposed scheme, the growth of the bank's database was successfully minimized and the issue of e-coins is controlled by several authorities. Our proposed scheme can greatly simplify the voting processes when several elections are to be held in a short period of time by embedding information about each election in a partially blind threshold signature. In our scheme, the size of a partially blind threshold signature is the same as that of an individual partially blind signature and the signature verification process is simplified by a group public key. The security of our scheme relies on the difficulty of computing discrete logarithm. © 1999 Elsevier Science B.V. All rights reserved.

Keywords: Partially blind signatures; Threshold signatures; Discrete logarithm; Secure; E-cash systems; Secure voting schemes; Privacy and security

1. Introduction

不可链接性：可以防止signer将 签名过程 和 签名 联系起来

The concept of blind signature was introduced by Chaum [1]. It is an interactive protocol which involves two kinds of participants, the signer and requesters. It allows a requester to obtain signatures on messages he provides to the signer without revealing these messages. A distinguishing property required by a typical blind signature scheme [1–5] is 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 is later made public. In a distributed environment, every signed blind message can be regarded as a fixed amount of electronic money in secure electronic payment systems [1, 6–9], or as a ticket in applications such as secret voting schemes [10–12]. The security of the blind signature schemes proposed in [1, 4] is based on the hardness of factorization [13] while the security of the schemes proposed in [2, 5] is based on the hardness of computing discrete logarithm [14–16].

A relevant type of signature scheme called group signatures was introduced in [17] and several improved solutions were proposed in [18–20]. The schemes in [17–20] allow a group member to sign a message on the group's behalf such that everybody can verify the signature but no one can find out which group member signed it. However, in case of a later dispute, a designated group manager can reveal the identity of the signer. These schemes can be generalized to allow defined subsets of all group members to jointly sign a message on behalf of the group. Group signatures can be used by a corporation for authenticating price lists or digital contracts. The customers need to know only a single group public key to verify signatures. The anonymity in schemes [17–20] is between the signature and signers who signed it, while the anonymity in schemes [1–5] is between the signature and the requester who provided the signed message.

Up to date, the on-line e-cash systems proposed in [1, 6] are quite efficient and practical. The aim of these systems was to produce an electronic version of money which retains the same properties as paper cash. These systems involve customers, the bank and the shops. In general, it is very hard to find a single entity which will be trusted by everyone else (such as the bank). To cope with this dilemma, two group-oriented blind threshold signature schemes [21] were proposed in a distributed environment, where several signers work together to sign a blind threshold e-coin. The schemes proposed in [21] allow t out of n participants in a group cooperating to sign a blind threshold signature. These schemes can be directly applied to the

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

PII: S0140-3664(98)00214-X

secure e-cash systems [1, 6] for distributing the power of a single authority. The modified e-cash systems can be used in the real world environments without a single trusted authority or with some absent/dishonest authorities. Another major problem in the on-line e-cash systems proposed in Refs. [1, 6] is the unlimited growth of the bank's database which keeps all used e-cashes for preventing double spending. To cope with this problem, the concept of partially blind signatures is introduced in Ref. [22].

In this paper, we propose a partially blind threshold signature scheme to address both the previous problems. In our scheme, the size of a partially blind threshold signature is the same as that of an individual partially blind signature and the verification process of a partially blind threshold signature is simplified by a group public key. The proposed scheme provides the message recovery capability [13, 23]. The security of our scheme 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.

The paper is organized as follows. In Section 2, we present a generalized partially blind signature scheme based on discrete logarithm, which will be used in our proposed partially blind threshold signature scheme. In Section 3, we present an efficient partially blind threshold signature scheme. Then we discuss its correctness, security, performance and extensions in Section 4. In Section 5, we describe some applications of the scheme. Finally, a concluding remark is given in Section 6.

问题1：找到一个可信的单一实体是困难的
问题2：数据库无限增长，因为需要保存所有签名数据

2. Partially blind signatures based on discrete logarithm

In this section, we propose a generalized partially blind signature scheme with message recovery. We first define the partial blindness of a digital signature scheme as follows:

Definition 1. A partially blind signature scheme with message space $M \subseteq \{0, 1\}^*$ is a 7-tuple $P = (\Psi, \Re, \partial, \Omega, \Upsilon, \Phi, \Gamma)$, where

- Ψ is the signer of the scheme;
- $\Re$ is a set of requesters;
- ∂ is a poly-time algorithm that on input a random string χ, constructs the signer's secret key (sk) and its corresponding public key (pk);
- Ω is a poly-time algorithm that on input a message $m \in M$, a negotiated constant c and a random string λ, constructs a partially blind message (m', c), 输入消息m，c，随机数λ 输出部分盲的消息(m',c)
- Υ is a poly-time algorithm that on input a partially blind message (m', c) and the secret key (sk), constructs a blind signature σ' on (m', c); 根据(m',c)构造一个盲签名
- Φ is a poly-time algorithm that on input a blind signature σ' and the random string λ, extracts the signature σ for (m, c);
- Γ is a poly-time algorithm that on input a message-signature pair $[(m, c), \sigma]$ and the public key (pk), outputs either yes or no,

such that, we have the following:

1. At the beginning of the signature generation, a requester $R \in \Re$ negotiates a constant c as the common information with the signer Ψ. 签名生成的初始阶段，R与signer会协商一个常量c作为公共参数
2. Ψ cannot embed any other extra information into the blind signature σ'. signer不能将任何信息嵌入签名
3. A poly-time bounded $R \in \Re$ cannot forge a signature with a different negotiated constant. R不可以用其他常量来构造一个签名
4. Under the same negotiated constant c, all signatures generated by the scheme are blind signatures.
在同一个c下，方案所产出的所有签名都是盲签名

In a typical signing process of a partially blind signature scheme, there are two kinds of participants, the signer and requesters. The proposed scheme consists of two phases: (1) the signature generation phase; and (2) the signature verification phase. In the signature generation phase, a requester requests a partially blind signature from the signer and the signer issues the partially blind signature to the requester. In the signature verification phase, anyone can use the public key to verify if a partially blind signature is valid. For simplicity, the partially blind signature scheme is based on the Nyberg-Rueppel blind signature scheme [2] with message recovery. All secure Meta-ElGamal blind signature schemes proposed in [2, 5] can be used in our scheme.

Let m be the blind message to be signed, h_1 and h_2 be two secure one-way hashing functions [24–26], p and q be two large strong prime numbers [16] such that q divides $(p - 1)$ and ρ be a generator of Z_p^* (i.e. $\gcd(\rho, p) = 1$, $\rho \neq 1$). Let $g \equiv_p \rho^{(p-1)/q}, z_1, z_2 \in Z_q$ be the signer's secret keys and $y_1 \equiv_p g^{z1}, y_2 \equiv_p g^{z_2}$ be the corresponding public keys. Let $\varpi_1(x) \equiv_q \sum_{1=0}^{\nu} \phi_i x^i$ and $\varpi_2(x) \equiv_q \sum_{i=0}^{\tau} \pi_i x^i$, where $\phi_i \in Z_q$ and $\pi_i \in Z_q$, be two non-zero public polynomials with degrees of at least 1. Let $\varpi(x, y, c) \equiv_q \varpi_1(h_1(c))x + \varpi_2(h_2(c))y$ and be a public polynomial.

2.1. The signature generation phase

When a requester requests a partially blind signature, he negotiates a constant c as the common information with the signer. At this time, $y_c \equiv_p g^{\varpi(z_1,z_2,c)} \equiv_p g^{\varpi_1(h_1(c))z_1+\varpi_2(h_2(c))z_2} \equiv_p y_1^{\varpi_1(h_1(c))} y_2^{\varpi_2(h_2(c))}$ is the signer's public key that contains the constant c and the coresponding secret key is $\varpi(z_1,z_2,c) \equiv_q \varpi_1(h_1(c))z_1 + \varpi_2(h_2(c))z_2$. Then they perform the following steps.

1. The signer randomly chooses a number $k \in Z_q$ computes $\hat{r} \equiv_p g^k$ and sends $\hat{r}$ to the requester.
2. After receiving $\hat{r}$, the requester does the following.
   1. (a) Choose two random numbers $\alpha \in Z_q$ and $\beta \in Z_q^*$, compute $r \equiv_p mg^{\alpha}\hat{r}^{\beta}$ and $\hat{m} \equiv_q \beta^{-1}r$.
   2. (b) Check if $\hat{\mathrm{m}} \neq 0$. If yes, send $\hat{m}$ to the signer. Otherwise, go back to step (a).
3. Upon receiving $\hat{m}$, the signer computes $\hat{s} \equiv_q \hat{m}\varpi(z_1,z_2,c) + k$ and sends $\hat{s}$ back to the requester.
4. After receiving $\hat{\mathrm{s}}$, the requester computes $s \equiv_q \hat{s}\beta + \mathrm{\alpha}$. The partially blind signature of m is (r, s, c).

2.2. The signature verification phase

To verify the signature (r, s, c), one simply computes $y_c \equiv_p y_1^{\varpi_1(h_1(c))} y_2^{\varpi_2(h_2(c))}$ and $m \equiv_p g^{-s}y_c^r r$ and checks if m has some redundancy information. If m has no proper redundancy, a secure one-way hashing function h [24–26] can be applied to m. But this approach can not provide the message recovery capability. To verify the partially blind signature (r, s, c) on m without redundancy, one must send m along with (r, s, c) to the verifier.

2.3. Analysis

Let $\nu = \{c, k, \hat{r} \equiv_p g^k, \hat{m}, \hat{s} \equiv_q \hat{m}\varpi(z_1,z_2,c) + k\}$ denote the signer's complete view of an execution in the signature generation phase and $[m, (r, s, c)]$ denote the message-signature pair generated in that execution. Theorem 1 ensures the correctness and blindness of the scheme.

Theorem 1. The 3-tuple (r, s, c) is a valid partially blind signature on message m for the Nyberg-Rueppel signature scheme and under the same negotiated constant c, our proposed scheme is a blind signature scheme.

Proof. The validity of the partially blind signature (r, s, c) on the message m can easily be established as follows.

$$g^{-s}y_c^r r$$

$$\equiv_p g^{-(\hat{s}\beta+\alpha)}(y_1^{\varpi_1(h_1(c))} y_2^{\varpi_2(h_2(c))})^r mg^{\alpha+k\beta}$$

$$\equiv_p g^{-((\hat{m}(\varpi(z_1,z_2,c))+k)\beta+\alpha)} g^{\varpi(z_1,z_2,c)} mg^{\alpha+k\beta}$$

$$\equiv_p g^{-\varpi(z_1,z_2,c)r-k\beta-\alpha} g^{\varpi(z_1,z_2,c)} mg^{\alpha+k\beta}$$

$$\equiv_p m$$

For proving the blindness of the scheme, we show that given any view ν and any valid message-signature pair $[m, (r, s, c)]$, there exists a unique pair of blinding factors α and β. Since the requester chooses the blinding factors α and β randomly, the blindness of the signature scheme follows.

Assume that the partially blind signature (r, s, c) was generated by the signer with the view consisting of $c, k, \hat{r} \equiv_p g^k, \hat{s} \equiv_q \hat{m}\varpi(z_1,z_2,c) + k$ and $\hat{m}$, then the following equations must hold for α and β.

$$r \equiv_p mg^{\alpha}\hat{r}^{\beta} \qquad (1)$$

$$\hat{m} \equiv_q r\beta^{-1} \qquad (2)$$

$$s \equiv_q \hat{s}\beta + \alpha \qquad (3)$$

Since $\hat{m} \in Z_q$ and $\hat{m} \neq 0$, by Eqs. (2) and (3), the unique so1ution for α and β is:

$$\beta \equiv_q r\hat{m} - 1 \tag{4}$$

$$\alpha \equiv_q s - \hat{s}\beta \tag{5}$$

In the following, we show that the solution of α and β in Eqs. (4) and (5) also satisfies Eq. (1).

$$mg^{\alpha}\hat{r}^{\beta}$$

$$\equiv_p g^{-s}y_c^{r}rg^{\alpha}g^{k\beta}$$

$$\equiv_p rg^{-(\hat{s}\beta+\alpha)}g^{\varpi(z_1,z_2,c)r}g^{\alpha}g^{k\beta}$$

$$\equiv_p rg^{-(\hat{m}\varpi(z_1,z_2,c)+k)\beta-\alpha}g^{\varpi(z_1,z_2,c)r}g^{\alpha}g^{k\beta}$$

$$\equiv_p r$$

□

Then we show that our proposed scheme satisfies the following two conditions: (1) the signer can not embed any other information into the partially blind signature to link this signing process with the message-signature pair published later; and (2) a poly-time bounded requester can not forge a signature with a different negotiated constant. First, condition (1) is discussed. Since the constant c is negotiated between the requester and the signer at the beginning of the signature generation phase, the partially blind signature can not pass the verification function if the signer adds any extra information in c. Then, we discuss how our scheme satisfies condition (2). Let (r, s, c) be the partially blind signature on a message m and S_q which is generated by a generator $g \equiv_p \rho^{(p-l)/q}, \rho \in Z_p^*$, be a cyclic subgroup of Z_p^* with $|S_q| = q$. Lemma 2 ensures that given a message-signature pair [m, (r, s, c)], to forge a partially blind signature (r, s_1, c_1) on the message m is equivalent to solve the discrete logarithm problem in a subgroup. Lemma 3 ensures that given a message-signature pair [m, (r, s, c)], to forge a partially blind signature (r_1, s, c_1) on the message m and a partially blind signature (rr_1, s, c_2) on the message m^2 is also equivalent to solve the discrete logarithm problem in a subgroup.

Let P1, P2 and P3 refer to the problems presented later and let O1, O2 and O3 be the oracles that solve these problems in the same order as given.

(P1) Given a message-signature pair [m, (r, s, c)] and the corresponding public key y_c, derive another partially blind signature (r, s_1, c_1) on the message m.

(P2) Given a message-signature pair [m, (r, s, c)] and the corresponding public key y_c, derive another partially blind signature r_1, s, c_1) on te message m and a partially blind signature (r_1r, s, c_2) on the message m^2

(P3) Given an input value $G \in S_q$, derive an integer k such that $G \equiv_p g^k$

Lemma 2. The difficulty of solving problem P1 is equivalent to that of solving problem P3.

Proof. It is obvious that oracle O3 is polynomial-time transformable to an algorithm for P1.

Now, we show that oracle O1 is polynomial-time transformable to an algorithm for P3.

We define an algorithm A for solving P3 using O1 as following.

```
Algorithm A($G \in S_q$)
    Step 1: randomly choose $r \in Z_q^*$ and $s \in Z_q$.
    Step 2: randomly choose a constant $c$, such that $\varpi_1(h(c)) \neq 0$ and $\varpi_2(h(c)) \neq 0$.
    Step 3: compute $m \equiv_p r(G)^{-1}$ and $y_c \equiv_p (g^sG^{-1})^{r^{-1}}$
    Step 4. query oracle O1 with input [$m$, ($r$, $s$, $c$), $y_c$]. Let ($r$, $s_1$, $c_1$) be the result.
    Step 5: query oracle O1 with input [$m$ ($r$, $s_1$, $c_1$), $y_c$]. Let ($r$, $s_2$, $c_2$) be the result.
    Step 6: solve the following three linear equations
```

$$s \equiv_q \varpi_1(h_1(c))z_1r + \varpi_2(h_2(c))z_2r + k \tag{6}$$

$$s_1 \equiv_q \varpi_1(h_1(c_1))z_1r + \varpi_2(h_2(c_1))z_2r + k \tag{7}$$

$$s_2 \equiv_q \varpi_1(h_1(c_2))z_1r + \varpi_2(h_2(c_2))z_2r + k \tag{8}$$

```
    and obtain $(z_1, z_2, k)$
    Step 7: output $k$.
end.
```

If $[m\ (r, s, c)]$ is a valid message-signature pair on the public key y_c, it must satisfy Eqs. (6) and (9).

$$r \equiv_p mg^k \tag{9}$$

The validity of the partially blind signature (r, s, c) on the message m can easily be established as follows. $g^{-s}y_c^r r \equiv_p g^{-s}((g^sG^{-1})^{r^{-1}})^r r \equiv_p g^{-s}g^sG^{-1}mg^k \equiv_p m.$

The partially blind signatures (r, s_1, c_1) and (r, s_2, c_2) on the message m are signed by the secret keys $\varpi(z_1, z_2, c_1)$, $\varpi(z_1, z_2, c_2)$ and Eqs. (7) and (8) must hold.

Since $\varpi_1(x)$ and $\varpi_2(x)$ are two public polynomials, h_1 and h_2 are two public one-way hashing functions, c, c_1, c_2, r, s, s_1 and s_2 are public values, the requester can solve linear Eqs. (6) and (7) to get a solution (z_1, z_2, k).

Clearly, the algorithm A can be constructed from O1 in polynomial time.□

Lemma 3. The difficulty of solving problem P2 is equivalent to that of solving problem P3.

Proof. It is obvious that oracle O3 is polynomial-time transformable to an algorithm for P2.

Now, we show that oracle O2 is polynomial-time transformable to an algorithm for P3.

We define an algorithm A for solving P3 using O2 as following.

```
Algorithm A($G \in S_q$)
Step 1: randomly choose $r \in Z_p^*$ and $s \in Z_q$.
Step 2: randomly choose a constant $c$, such that $\varpi_1(h(c)) \neq 0$ and $\varpi_2(h(c)) \neq 0$.
Step 3: compute $m \equiv_p r(G)^{-1}$ and $y_c \equiv_p (g^sG^{-1})^{r^{-1}}$
Step 4: query oracle O2 with input $[m, (r, s, c), y_c]$. Let $(r_1, s, c_1)$ and $(r_1r, s, c_2)$ be the result.
Step 5: query oracle O2 with input $[m, (r_1, s, c_1), y_c]$. Let $(r_2, s, c_3)$ and $(r_1r_2, s, c_4)$ be the result.
Step 6: solve the following five linear equations
```

$$s \equiv_q \varpi_1(h_1(c))z_1r + \varpi_2(h_2(c))z_2r + k \tag{10}$$

$$s \equiv_q \varpi_1(h_1(c_1))z_1r_1 + \varpi_2(h_2(c_1))z_2r_1 + k_1 \tag{11}$$

$$s \equiv_q \varpi_1(h_1(c_2))z_1rr_1 + \varpi_2(h_2(c_2))z_2rr_1 + (k_1 + k) \tag{12}$$

$$s \equiv_q \varpi_1(h_1(c_3))z_1r_2 + \varpi_2(h_2(c_3))z_2r_2 + k_2 \tag{13}$$

$$s \equiv_q \varpi_1(h_1(c_4))z_1r_1r_2 + \varpi_2(h_2(c_4))z_2r_1r_2 + (k_1 + k_2) \tag{14}$$

```
and obtain $(z_1, z_2, k, k_1, k_2)$
Step 7: output k
end.
```

□

Without any valid message-signature pair, the hardness of faking a valid partially blind signature (r_1, s_1, c_1) on message m in our scheme is similar to the Nyberg-Rueppel signature scheme since given the message m with redundancy and the value r_1, one has to derive the corresponding value s_1 by solving Eq. (15). To solve this equation, one has to solve the discrete logarithm problem in a subgroup.

$$m \equiv_p g^{-s_1} y_{c_1}^{r_1} r_1 \tag{15}$$

In order to verify the partially blind signature (r, s, c), a verifier has to compute the new public key y_c from all public values y, g, c. It is difficult to compute the new public key y_c from the public values $g, y(\equiv {}_p g^z)$ without first obtaining the secret key z when the new secret key z_c, contains z^ϵ, where $\epsilon \in Z$ and $\epsilon \geq 2$. The simple case that $y_c \equiv_p g^{z^2}$ is a special case that each user chooses the same secret key z, sends $Y \equiv_p g^z$ to each other and computes the common key $K_{ij} \equiv_p Y^z \equiv g^{z^2}$ in the Diffle-Hellman key-exchange protocol [27].

In our proposed partially blind signature scheme, the negotiated constant c is embedded into the new secret key $z_c = \varpi(z_1, z_2, c) \equiv_q \varpi_1(h_1(c))z_1 + \varpi_2(h_2(c))z_2$, which is the linear combination of two secret keys z_1 and z_2. This realization can prevent the requester from forging another negotiated constant c' by sending another blind message $\hat{m}' \equiv {}_p \xi \hat{m}$, where ξ is a combination of all known values, to the signer in step 2 of the signature generation phase. Simply using a single secret key z in our partially blind signature scheme will allow the requester to forge another fake constant c' by sending another blind message to the signer.

For example, let $z_c = \varpi(z, c) \equiv_q \varpi_1(h_1(c))z + \varpi_2(h_2(c))$ and the corresponding public key is $y_c =_p y^{\varpi_1(h_1(c))} g^{\varpi_2(h_2(c))}$, which can be easily computed by anyone. For forging a fake partially blind signature (r, s', c') instead of the partially blind signature (r, s, c) on a message m, the requester can send the blind message:

$$\hat{m}' \equiv_p \frac{\varpi(h_1(c'))}{\varpi_1(h_1(c))} \hat{m}$$

to the signer in step 2 of the signature generation phase. In step 4, the requester can derive the fake signature (s', r, c') by the following equation:

$$s' \equiv_q \hat{s}\beta + \alpha + r\left(\varpi_2(h_2(c')) - \varpi_2(h(c)) \frac{\varpi_1(h_1(c'))}{\varpi_1(h_1(c))} \right)$$

$$\equiv_q (\hat{m}' \varpi(z, c) + k)\beta + \alpha + r\left(\varpi_2(h_2(c')) - (\varpi_2(c)) \frac{\varpi_1(h_1(c'))}{\varpi_1(h_1(c))} \right)$$

$$\equiv_q \beta^{-1} r \frac{\varpi_1(h_1(c'))}{\varpi_1(h_1(c))} \left(\varpi_1(h_1(c))z + \varpi_2(h_2(c)))\beta + k\beta + \alpha + r(\varpi_2(h_2(c')) - \varpi_2(h(c)) \frac{\varpi_1(h_1(c'))}{\varpi_1(h_1(c))} \right)$$

$$\equiv_q r(\varpi_1(h_1(c'))z + \varpi_2(h_2(c'))) + k\beta + \alpha$$

$$\equiv_q r(\varpi(z, c') + k\beta + \alpha$$

Now we have

$$g^{s'} y_{c'}^{-r} r$$

$$\equiv_p g^{r(\varpi(z,c')) + \beta k + \alpha} g^{\varpi(z,c')(-r)} m g^{\beta k + \alpha} \equiv_p m$$

With the information of the partially blind signature, an attacker is not capable of deriving the secret keys z_1 and z_2 since it needs to solve the equation $m \equiv_p g^{-s} y_c^r r \equiv_p g^{-s} (g^{\varpi_1(h(c))z_1 + \varpi_2(h(c))z_2})^r r$. To solve this equation, one has to solve the discrete logarithm problem in a subgroup.

3. The partially blind threshold signature scheme

In this section, we propose a partially blind threshold signature scheme with message recovery. We first define the partial blindness of a threshold signature scheme as follows:

Definition 2. A partially blind threshold signature scheme is a threshold signature scheme such that: (a) this scheme involves n signers and a set of requesters; (b) at the beginning of the signature generation, a requester negotiates a constant c as the common information with t out of n signers; (c) the negotiated constant c and the blind message m provided by the requester will be signed together by the t signer; (d) under the same negotiated constant c, all threshold signatures generated by the scheme are blind threshold signatures; (e) any signer cannot embed any other extra information into the threshold signature; and (f) a poly-time bounded requester cannot forge a signature with a different negotiated constant.

In a typical signing process of a partially blind threshold signature schemes, there are two kinds of participants, the signers and a requester. Before the requester can obtain a partially 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 partially 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 partially blind threshold signature from the signers and the signers cooperate to issue the partially blind threshold signature to the requester. In the signature verification phase, anyone can use the group public key to verify if a partially 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 partially blind threshold signature scheme, m be the blind message to be signed, h, h_1 and h_2 be three secure one-way hashing functions [24–26], p and q be two large strong prime numbers such that q divides $(p - 1)$ and p be a generator of Z_p^* (i.e. gcd(ρ,p) = 1, ρ ≠ 1). Let $g \equiv_p \rho^{(p-1)/q}$. 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 [28]). Using a secure public key signature scheme [13–15], U_i can produce signatures (certificates) of messages by his own secret key d_i. Anyone can verify these signatures by the corresponding public key e_i. Let $Cert_{U_i}(m)$ be the signature (certificate) on the message m produced by U_i and ϖ_1 (x), ϖ_2 (x) and ϖ (x, y, z) be defined in Section 2.

3.1. The shadow distribution phase

Before a requester can request a partially 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 $z_{i,1}$, $z_{i,2} \in Z_q$ and two secret polynomials $f_{i,1}(x) = \sum_{k=0}^{t-1} a_{i,k}x^k, f_{i,2}(x) = \sum_{k=0}^{t-1} b_{i,k}x^k$ such that $a_{i,0} = z_{i,1}, b_{i,0} = z_{i,2}$ and $a_{i,j} \in Z_q, b_{i,j} \in Z_q, 1 \le j \le t-1$, computes $\Psi_{i,k,1} \equiv_p g^{a_{i,k}}$, $\Psi_{i,k,2} \equiv_p g^{b_{i,k}}$ and signatures $Cert_{U_i}(h(\Psi_{j,k,1})), Cert_{U_i}(h(\Psi_{j,k,2}))$ and sends $((Cert_{U_i}(h(\Psi_{j,k,1}))$, $Cert_{U_i}(h(\Psi_{j,k,2}))$, $\Psi_{i,k,1}, \Psi_{i,k,2}), 0 \le k \le t-1)$ to $U_j, 1 \le j \le n, j \ne i$.

2. Upon receiving $((Cert_{U_j}(h(\Psi_{j,k,1})), Cert_{U_j}\ (h(\Psi_{j,k,2})), \Psi_{i,k,1}, \Psi_{i,k,2}), 1 \le j \le n, j \ne i, 0 \le k \le t-1)$ from all other signers, U_i verifies if all $((Cert_{U_j}\ (h(\Psi_{j,k,1})), Cert_{U_j}(h(\Psi_{j,k,2}))$ are valid. If yes, he sends $\delta_{i,j,1} \equiv_q f_{i,1}(x_j)$, $\delta_{i,j,2} \equiv_q f_{i,2}(x_j)$, where x_j is a unique public number for U_j, and the two corresponding signatures $Cert_{U_i}(h(\delta_{i,j,1})), Cert_{U_i}(h(\delta_{i,j,2}))$ 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 $\delta_{j,i,1}$, $\delta_{j,i,2}$, $Cert_{U_j}\ (h(\delta_{j,i,1})), Cert_{U_j}(h(\delta_{j,i,2}))1 \le j \le n, j \ne i$ from other signers, he verifies if the shares $\delta_{j,i,1}, \delta_{j,i,2}$, received from U_j is consistent with the certified values $\Psi_{j,l,1}$, $\Psi_{j,l,2}0 \le l \le t-1$, by checking whether $g^{\delta_{j,i,1}} \equiv_p \Pi_{l=0}^{t-1}(\Psi_{j,l,1})^{x_i^l}$ and $g^{\delta_{j,i,2}} \equiv_p \Pi_{l=0}^{t-1}(\Psi_{j,l,2})^{x_i^l}$. If it fails, U_i broadcasts that an error was found, publishes $\delta_{j,i,1}, \delta_{j,i,2}, Cert_{U_j}(h(\delta_{j,i,1})), Cert_{U_j}(h(\delta_{j,i,2}))$ 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 $y \equiv_p \Pi_{l=1}^{n} y_{l},_1\ Y_{l,2} \equiv_p \Pi_{l=1}^{n} \Psi_{l,0,1}\Psi_{l,0,2}$ the public shadows $\Phi_{j,i,1} \equiv_p g^{\delta_{j,i,1}}$, $\Phi_{j,i,2} \equiv_p g^{\delta_{j,i,2}}$ and the signatures $Cert_{U_i}(h(\Phi_{j,i,1})), Cert_{U_i}(h(\Phi_{j,i,2})), 1 \le j \le n$. He then sends $(Cert_{U_i}(h(y))$, $(\Phi_{j,i,1}, \Phi_{j,i,2}\ Cert_{U_i}(h(\Phi_{j,i,1}))$, $Cert_{U_i}(h(\Phi_{j,i,2}))$, $1 \le j \le n))$ to all other signers.

4. Upon receiving all $((Cert_{U_j}(h(y))$, $1 \le j \le n, j \ne i), (\Phi_{l,j,1}$, $\Phi_{l,j,2}$, $Cert_{U_j}(h(\Phi_{l,j,1}))$, $Cert_{U_j}(h(\Phi_{l,j,2}))$, $1 \le l \le n, 1 \le j \le n, j \ne i))$, U_i verifies if all $((Cert_{U_j}(h(y))$, $1 \le j \le n, j \ne i)$, $(Cert_{U_j}(h(\Phi_{l,j,1}))$, $(Cert_{U_j}(h(\Phi_{l,j,2}))$, $1 \le l \le n, 1 \le j \le i, j \ne i))$ are valid. If yes, the shadow keys corresponding to the group secret key $z \equiv_q \sum_{j=1}^{n}(z_{j,1} + z_{j,2})$were securely and correctly distributed. The group public key $y \equiv_p \Pi_{j=1}^{n} y_{j,1}y_{j,2} \equiv_p g^{\Sigma_{j=1}^{n}(z_{j,1}+z_{j,2})}$, all signers public keys $y_{j,1} \equiv_p \Psi_{j,0,1}, y_{j,2} \equiv_p \Psi_{j,0,2},, 1 \le j \le n$, and all public shadows $\Phi_{l,j,1} \equiv_p g^{\delta_{l,j,1}}$, $\Phi_{l,j,2} \equiv_p g^{\delta_{l,j,2}}$, $1 \le l, j \le n$ 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, $1 \le i \le t$. When a requester requests a partially blind threshold signature, he negotiates a constant c as the common information with the t signers. At this time, $y_{c,i} \equiv_p g^{\varpi(z_{i,1}, z_{2,2}, c)} \equiv_p y_{i,1}^{\varpi(h_1(c))} y_{i,2}^{\varpi(h_2(c))}$ is U_i's public key that contains the negotiated constant c and the corresponding secret key is

$z_{c,i} = \varpi(z_{i,1}, z_{i,2}, c) = \varpi_1(h_1(c))z_{i,1} + \varpi_2(h_2(c))z_{i,2}$. The group public key is $y_c \equiv_p \Pi_{i=1}^{n} y_{c,i} \equiv_p \Pi_{i=1}^{n}(y_{i,1}^{\varpi_1(h_1(c))} y_{i,2}^{\varpi_2(h_2(c))})$ Then they perform the following steps during the signature generation phase.

1. Each U_i randomly chooses a number $k_i \in Z_q$, computes $\hat{r}_i \equiv_p g^{k_i}$ and sends $\hat{r}_i$ to the requester.
2. After receiving all $\hat{r}_i, 1 \le i \le t$, the requester does the following:

   (a) choose two random numbers$\alpha \in Z_q$ and $\beta \in Z_q^*$, compute $r_i \equiv_p g^{\alpha}\hat{r}_i^{\beta}, r \equiv_p m\Pi_{i=1}^{t} r_i$ and $\hat{m} \equiv_q \beta^{-1} r$;
   (b) check if $\hat{m} \neq 0$. If yes, send $\hat{m}$ to all$U_i, 1 \le i \le t$. Otherwise, go back to step (a).

3. Upon receiving $\hat{m}$, each U_i computes:

$$\hat{s}_i \equiv_q \hat{m}\left(\varpi(z_{i,1}, z_{i,2}, c) + \varpi_1(h_1(c)) \sum_{j=t+1}^{n} \left(f_{j,1}(x_i)\left(\prod_{k=1,k\neq i}^{t}\left(\frac{-x_k}{x_i - x_k}\right)\right)\right) + \varpi_2(h_2(c)) \sum_{j=t+1}^{n} \left(f_{j,2}(x_i)\left(\prod_{k=1,k\neq i}^{t}\left(\frac{-x_k}{x_i - x_k}\right)\right)\right)\right) + k_i$$

   and sends $\hat{s}_i$ back to the requester.
4. After receiving all $\hat{s}_i$, the requester computes $s_i \equiv_q \hat{s}\beta + \alpha$ and checks if:

$$g^{-s_i} y_{c,i}^{r} r_i \equiv_p \left(\prod_{j=t+1}^{n}(\Phi_{j,i,1})\right)^{\left(\prod_{k=1,k\neq i}^{t}\left(\frac{-x_k}{x_i - x_k}\right)\right)(-r\varpi_1(h_1(c)))} \left(\prod_{j=t+1}^{n}(\Phi_{j,i,2})\right)^{\left(\prod_{k=1,k\neq i}^{t}\left(\frac{-x_k}{x_i - x_k}\right)\right)(-r\varpi_2(h_2(c)))} ,1 \le i \le t\ .$$

   If $\hat{s}_i$ 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 partially blind threshold signature of m is (r, s, c).

3.3. The signature verification phase

To verify the partially blind threshold signature (r, s, c), one simply computes $y_c \equiv_p \Pi_{i=1}^{n} y_{c,i} \equiv_p \Pi_{i=1}^{n}(y_{i,1}^{\varpi_1(h_1(c))} y_{i,2}^{\varpi_2(h_2(c))})$ and $m \equiv_p g^{-s}(y_c)^r r$ and checks if m has some redundancy information. If m has no proper redundancy, a secure one-way hashing function h [24–26] can be applied to m. But this approach can not provide the message recovery capability. To verify the partially blind threshold signature (r, s, c) on m without redundancy, one must send m along with (r, s, c) to the verifier.

4. Discussion

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

4.1. Correctness

Let:

$$\upsilon = \{c, k_i, \hat{r}_i \equiv_p g^{k_i}, \hat{m}, \hat{s}_i \equiv_q \hat{m}\left(\varpi(z_{i,1}, z_{i,2}, c) + \varpi_1(h_1(c)) \sum_{j=t+1}^{n} \left(f_{j,1}(x_i)\left(\prod_{k=1,k\neq i}^{t}\left(\frac{-x_k}{x_i - x_k}\right)\right)\right) + \varpi_2(h_2(c))\right.$$
$$\left.\times \sum_{j=t+1}^{n} \left(f_{j,2}(x_i)\left(\prod_{k=1,k\neq i}^{t}\left(\frac{-x_k}{x_i - x_k}\right)\right)\right)\right) + k_i, 1 \le i \le t\}$$

denote the signers complete view of an execution in the signature generation phase and [m (r, s, c)] denote the message-signature pair generated in that execution. 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 4. The partial signature (r_i, s_i, c) is valid if U_i is honest.

Proof. By our scheme, we have:

$g^{-s_i} y_{c,i}^{r} r_i$

$$\equiv_p g^{-(\hat{s}_i\beta+\alpha)}g^{\varpi(z_{i,1},z_{i,2},c)r}g^{\alpha}\hat{r}_i^{\beta}$$

$$\equiv_p g^{-\left(\hat{m}\left(\varpi(z_{i,1},z_{i,2},c)+\varpi_1(h_1(c))\sum_{j=t+1}^{n}f_{j,1}(x_i)\left(\prod_{k=1,k\neq i}^{t}\left(\frac{-x_k}{x_i-x_k}\right)\right)+\varpi_2(h_2(c))\sum_{j=t+1}^{n}f_{i,2}(x_i)\left(\prod_{k=1,k\neq i}^{t}\left(\frac{-x_k}{x_i-x_k}\right)\right)\right)+k_i\right)\beta}g^{\varpi(z_{i,1},z_{i,2},c)r}g^{k_i\beta}$$

$$\equiv_p g^{\left(-\hat{m}\varpi(z_{i,1},z_{i,2},c)-\hat{m}\varpi_1(h_1(c))\sum_{j=t+1}^{n}f_{j,1}(x_i)\left(\prod_{k=1,k\neq i}^{t}\left(\frac{-x_k}{x_i-x_k}\right)\right)-\hat{m}\varpi_2(h_2(c))\sum_{j=t+1}^{n}f_{j,2}(x_i)\left(\prod_{k=1,k\neq i}^{t}\left(\frac{-x_k}{x_i-x_k}\right)\right)\right)\beta}g^{\varpi(z_{i,1},z_{i,2},c)r}$$

$$\equiv_p g^{\left(\varpi_1(h_1(c))\sum_{j=t+1}^{n}f_{i,1}(x_i)\left(\prod_{k=1,k\neq 1}^{t}\left(\frac{-x_k}{x_i-x_k}\right)\right)+\varpi_2(h_2(c))\sum_{j=t+1}^{n}f_{i,2}(x_i)\left(\prod_{k=1,k\neq i}^{t}\left(\frac{-x_k}{x_i-x_k}\right)\right)\right)(-\hat{m}\beta)}$$

$$\equiv_p \left(\prod_{j=t+1}^{n}(\Phi_{j,i,1})\right)^{\left(\prod_{k=1,k\neq i}^{t}\left(\frac{-x_k}{x_i-x_k}\right)\right)-r\varpi_1(h_1(c)))}\left(\prod_{j=t+1}^{n}(\Phi_{j,i,2})\right)^{\left(\prod_{k=1,k\neq i}^{t}\left(\frac{-x_k}{x_i-x_k}\right)\right)(-r\varpi_2(h_2(c)))}$$

□

After the signature generation phase, the partially blind threshold signatures can be verified by the group public key in the signature verification phase. Lemma 5 ensures the correctness of the scheme.

Lemma 5. The 3-tuple (r, s, c) is a valid partially blind threshold signature on message m for the Nyberg–Rueppel signature scheme.

Proof. The validity of the partially blind threshold signature (r, s, c) on the message m can easily be established as follows.

$$g^{-s}y_c^r r \equiv_p g^{-\left(\sum_{i=1}^{t}(\hat{s}_i\beta+\alpha)\right)}g^{\left(\sum_{i=1}^{n}\varpi(z_{i,1},z_{,i,2},c)\right)r}\left(m\prod_{i=1}^{t}r_i\right)$$

$$\equiv_p mg^{-\sum_{i=1}^{t}\left(\hat{m}\left(\varpi(z_{i,1},z_{i,2},c)+\varpi_1(h_1(c))\sum_{j=t+1}^{n}\left(f_{i,1}(x_i)\left(\prod_{k=1,k\neq i}^{t}\left(\frac{-x_k}{x_i-x_k}\right)\right)\right)+\varpi_2(h_2(c))\sum_{j=t+1}^{n}\left(f_{j,2}(x_i)\left(\prod_{k=1,k\neq i}^{t}\left(\frac{-x_k}{x_i-x_k}\right)\right)\right)\right)+k_i\right)\beta-t\alpha}$$

$$\times g^{\left(\sum_{i=1}^{n}\varpi(z_{i,1},z_{i,2},c)\right)r}\left(\prod_{i=1}^{t}g^{\alpha}\hat{r}_i^{\beta}\right)$$

$$\equiv_p mg^{-\hat{m}\beta((\sum_{i=1}^{t}\varpi(z_{i,1},z_{i,2},c)+\varpi_1(h_1(c))\sum_{j=t+1}^{n}z_{i,1}+\varpi_2(h_2(c))\sum_{j=t+1}^{n}z_{i,2}}g^{\left(\sum_{i=1}^{n}\varpi(z_{i,1},z_{i,2},c)\right)r}$$

$$\equiv_p mg^{-r\left(\sum_{i=1}^{n}\varpi(z_{i,1},z_{i,2},c)\right)}g^{\left(\sum_{i=1}^{n}\varpi(z_{i,1},z_{i,2},c)\right)r}$$

$$\equiv_p m$$

□

4.2. Security analysis

Theorem 6. Under the same negotiated constant c, our proposed scheme is a blind threshold signature scheme.

Proof. For proving the blindness of the protocol, we show that given any view ν and a valid message-signature pair $[m, (r, s, c)]$, there exists a unique pair of blinding factors α and β. Since the requester chooses the blinding factors α and β randomly, the blindness of the signature scheme follows.

Without loss of generality, assume that the partially blind signature (r, s, c) was generated by t signers $U_i, 1 < i < t$, with the view ν consisting of:

$$c,\ k_i,\ \hat{r}_i \equiv_p g^{k_i},$$

$$\hat{s}_i \equiv_q \hat{m}\Bigg(\varpi(z_{i,1}, z_{i,2}, c) + \varpi_1(h_1(c)) \sum_{j=t+1}^{n} \Bigg(f_{j,1}(x_i)\Bigg(\prod_{k=1,k\neq i}^{t}\left(\frac{-x_k}{x_i - x_k}\right)\Bigg)\Bigg) + \varpi_2(h_2(c)) \sum_{j=t+1}^{n} \Bigg(f_{i,2}(x_i)\Bigg(\prod_{k=1,k\neq i}^{t}\left(\frac{-x_i}{x_i - x_k}\right)\Bigg)\Bigg)\Bigg) + k_i,$$

$1 \leq i \leq t$ and $\hat{m}$

then the following equations must hold for α and β.

$$r \equiv_p m \prod_{i=1}^{t} r_i \equiv_p m \prod_{i=1}^{t} g^{\alpha} \hat{r}_i^{\beta} \tag{16}$$

$$\hat{m} \equiv_q r\beta^{-1} \tag{17}$$

$$s \equiv_q \sum_{i=1}^{t} s_i \equiv_q \sum_{i=1}^{t} (\hat{s}_i\beta + \alpha) \tag{18}$$

Note that if $t < q$, then $\gcd(t, q) = 1$. Since $\hat{m} \in Z_q$ and $\hat{m} \neq 0$, by Eqs. (17) and (18), the unique solution for α and β is:

$$\beta \equiv_q\ r\hat{m}^{-1} \tag{19}$$

$$\alpha \equiv_q \left(s - \sum_{i=1}^{t} \hat{s}_i\beta\right) t^{-1} \tag{20}$$

In the following, we show that the solutions of α and β in Eqs. (19) and (20) also satisfies Eq. (16).

$$m \prod_{i=1}^{t} g^{\alpha} \hat{r}_i^{\beta}$$

$$\equiv_p g^{-s}(y_c)^r r g^{t\alpha} \prod_{i=1}^{t} g^{k_i\beta}$$

$$\equiv_p rg^{-\sum_{i=1}^{t}(\hat{s}_i\beta+\alpha)} g^{\left(\sum_{i=1}^{n}\varpi_1(z_{i,1},z_{i,2},c)\right)r} g^{t\alpha} g^{\beta\sum_{i=1}^{t}k_i}$$

$$\equiv_p rg^{-\Bigg(\hat{m}\Bigg(\sum_{i=1}^{t}(\varpi(z_{i,1},z_{i,2},c)+\varpi_1(h_1(c))\sum_{j=t+1}^{n}\Bigg(f_{i,1}(x_i)\Bigg(\prod_{k=1,k\neq i}^{t}\Bigg(\frac{-x_k}{x_i-x_k}\Bigg)\Bigg)\Bigg)\Bigg)+\varpi_2(h_2(c))\sum_{j=t+1}^{n}\Bigg(f_{i,2}(x_i)\Bigg(\prod_{k=1,k\neq i}^{t}\Bigg(\frac{-x_k}{x_i-x_k}\Bigg)\Bigg)\Bigg)\Bigg)+\sum_{i=1}^{t}k_i\Bigg)\beta-t\alpha}$$

$$\times g^{\left(\sum_{i=1}^{n}\varpi(z_{i,1},z_{i,2},c)\right)r} g^{t\alpha} g^{\beta\sum_{i=1}^{t}k_i}$$

$$\equiv_p rg^{-\left(\hat{m}\left(\sum_{i=1}^{t}\varpi(z_{i,1},z_{i,2},c)\right)+\varpi_1(h_1(c))\sum_{j=t+1}^{n}z_{i,1}+\varpi_2(h_2(c))\sum_{j=t+1}^{n}z_{i,2}\right)\beta} g^{\left(\sum_{i=1}^{n}\varpi(z_{i,1},z_{i,2},c)\right)r}$$

$$\equiv_p rg^{-r\left(\sum_{i=1}^{n}\varpi(z_{i,1},z_{i,2},c)\right)}g^{\left(\sum_{i=1}^{n}\varpi(z_{i,1},z_{i,2},c)\right)r}$$

$$\equiv_p r\ .$$

□

Given the secret information of a group of $l < t$ members, Lemma 7 ensures that the threshold cryptosystem constructed in the shadow distribution phase will not disclose any extra information about the group secret key $\sum_{i=1}^{n} z_{i,1} + z_{i,2}$.

Lemma 7. Given a group of $\sigma < t$ members $G = \{p_i | p_i \in [1,n], 1 \le i \le \sigma\}$ and the set of shares $\{\delta_{j,i,1}, \delta_{j,i,2} | 1 \le j \le n, i \in G\}$. For any fixed j, $1 \le j \le n$, it takes polynomial time on $|p|$ to generate two random set $\{g^{\widehat{a_{j,k}}} | 1 \le k \le t-1\}$ and $\{g^{\widehat{b_{j,k}}} | 1 \le k \le t-1\}$ satisfying $g^{\delta_{j,i,1}} \equiv_p \Pi_{k=0}^{t-1}(g^{\widehat{a_{j,k}}})^{x_i^k}$ and $g^{\delta_{j,i,2}} \equiv_p \Pi_{k=0}^{t-1}(g^{\widehat{b_{j,k}}})^{x_i^k}$ for $i \in G$.

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

$$g^{\delta_{j,i,1}} \equiv_p \prod_{l=0}^{t-1}(g^{a_{j,l,1}})^{x_i^l} \equiv_p g^{\sum_{l=0}^{t-1}\alpha_{j,l}*x_i^l} \tag{21}$$

Since $g \equiv_p \rho^{(p-1)/q}$ and ρ is a generator of Z_p^*, g generates a cyclic subgroup S_q of $Z_p^*$with $|S_q| = q$. From Eq. (21), we have

$$\delta_{j,i,1} \equiv_p \sum_{l=0}^{t-1} a_{j,l} * x_i^l \tag{22}$$

From (22), we know that given a fixed index j, the shares $\delta_{j,i,1}, i \in G$, will use the same variables, $\widehat{\alpha_{j,k}}$, $0 \le k \le t-1$, as follows:

$$\delta_{j,i,1} \equiv_q \sum_{k=0}^{t-1} \widehat{\alpha_{j,k}} * x_i^k \tag{23}$$

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

$$\delta_{j,i,1} \equiv_q \sum_{k=0}^{t-1} \widehat{a_{j,k}} * x_i^k (i \in G) \tag{24}$$

Since the linear equations have at least one solution $\widehat{a_{j,k}} = a_{j,k}, 0 \le k \le t-1$, we can solve the linear Eq. (24) and get a random solution $\widehat{\alpha_{j,ik}}$, $1 \le k \le t-1$, by assigning random values to all free variables. From Eq. (24), it is clear that

$$g^{\delta_{j,i,1}} \equiv_p g^{\sum_{k=0}^{t-1}\widehat{a_{j,k}}*x_i^k} \equiv \Pi_{k=0}^{t-1}(g^{\widehat{a_{j,k}}})^{x_i^k}$$

Similar to the previous proof, we can get a random solution $\widehat{b_{j,k}}, 1 \le k \le t-1$, such that

$$g^{\delta_{j,i,2}} \equiv_p g^{\sum_{k=0}^{t-1}\widehat{b_{j,k}},*x_i^k} \equiv_p \Pi_{k=0}^{t-1}(g^{\widehat{b_{j,k}}})^{x_i^k}$$

□

In our partially blind threshold signature scheme, the partial signature (s_i, r_i, c) must satisfy the equation

$$g^{-s_i}(y_{c,i})^r r_i \equiv_p g^{-s_i} g^{\varpi(z_{i,1},z_{i,2},c)r} r_i \equiv_p g^{-s_i} g^{(\varpi_1(h_1(c))z_{i,1} + \varpi_2(h_2(c))z_{i,2})r} r_i$$

$$\equiv_p \left(\prod_{j=t+1}^{n}(\Phi_{j,i,1})\right)^{\left(\prod_{k=1,k\neq i}^{t}\left(\frac{-x_i}{x_i-x_k}\right)\right)(-r\varpi_1(h_1(c)))}\left(\prod_{j=t+1}^{n}(\Phi_{j,i,2})\right)^{\left(\prod_{k=1,k\neq i}^{t}\left(\frac{-x_i}{x_i-x_k}\right)\right)(-r\varpi_2(h_2(c)))}$$

Since r, $\Phi_{j,i,1}$, $\Phi_{j,i,2}$, $\varpi_1(h_1(c))$, $\varpi_2(h_2(c)), x_k, r_i, y_i$ and s_i are all public, an attacker has to solve the discrete logarithm problem in order to get the secret values $z_{i,1}$ and $z_{i,2}$.

With the information of all partial signatures and the corresponding partially blind threshold signature, an attacker is not capable of deriving the secret keys $z_{i,1}$, $z_{i,2}$, $1 \le i \le n$, since he has to solve the equation

$$rg^{-s}y_c^r \equiv_p m(\Pi_{i=1}^n r_i)g^{-\left(\sum_{i=1}^n s_i\right)}(\Pi_{i=1}^n(g^{\varpi_1(h_1(c))z_{i,1}}g^{\varpi_2(h_2(c))z_{i,2}}))^r.$$

To solve this equation, one has to solve the discrete logarithm problem.

4.3. Performance analysis

In our scheme, the size of a partially blind threshold signature is the same as that of an individual partially blind signature and the verification process of a partially blind threshold signature is simplified by a group public key.

The major difference between RSA blind signature schemes and the partially blind signature scheme proposed in [22] is that the later embeds a negotiated constant c to the corresponding secret key d. This modification is simple and can be replaced by simply changing the secret key when the negotiated constant c is changed and authenticating the corresponding public key. In partially blind threshold signature schemes, this approach is very impractical since it has to redistribute the secret shadows, which is time consuming, when the negotiated constant c is changed. In our proposed scheme, the shadow distribution phase is done only once and then any t out of n signers can use the secret shadows to issue partially blind threshold signatures since when the constant c is negotiated, the public key $y_c \equiv_p \Pi_{i=1}^n y_{c,i} \equiv_p \Pi_{i=1}^n(y_{i,1}^{\varpi_1(h_1(c))}y_{i,2}^{\varpi_2(h_2(c))})$ can be computed by anybody since all $y_{i,1}, y_{i,2}, 1 \le i \le n, h_1, h_2$ and c are public and both $\varpi_1(x) \equiv_p \sum_{i=0}^{\nu} \phi_i x^i$ and $\varpi_2(x) \equiv_p \sum_{i=0}^{\tau} \pi_i x^i$ are public polynomials. For simplifying the verification process, each used public key y_c, can be recorded in a public key database. When a partially blind threshold signature is verified, the verifier can first retrieve the public key y_c from the database and then checks if this signature is valid. In this approach, the verification process of a partially blind threshold signature is equivalent to that of an individual signature. Also, in our scheme, the size of a partially blind threshold signature is the same as that of an individual partially blind signature. Thus, our proposed scheme is optimal with respect to the partially blind threshold signature size and the verification process.

4.4. Extensions

In [5], some extensions of the blind signature schemes in [2] were introduced. As mentioned in [5], not all variants of meta-message recovery signature schemes can be transformed to blind signature schemes. For examp1e, there is no blind signature scheme for the original ElGamal signature scheme yet. All extensions proposed in [5] except that $\tilde{B}$ contains $\tilde{s}$ in the signature generation equation, can be adopted into our proposed scheme. The frameworks of these extension schemes are similar to that of our proposed scheme. The security considerations and performance analysis of these extended schemes are similar to those of our proposed schemes.

5. Applications

5.1. Secure and efficient on-line e-cash systems

Up to date, the on-line e-cash systems proposed by Chaum [1, 6] are quite efficient and practical. These systems involve customers, the bank and the shops. In these systems, the protocols can be simplified as the following phases: the withdrawal phase, the spending phase and the deposit phase. During the withdrawal phase, customers apply the blind signature technique to get their blind e-cashes. In the spending phase, customers first generate their real e-cashes from the blind e-cashes received in the withdrawal phase and then spend them at the designated shops. Finally, in the deposit phase, the shops deposit the-e-cashes at the bank. The bank will check if the e-coins were used. In real world environments, it is very hard to find any single entity which will be trusted by everyone else (such as the bank) to issue e-cashes. To deal with the problem, some modifications of the schemes in [1, 6] must be made.

The modifications are as follows:

1. Instead of a unique authority, the modified systems consist of n administrators who will cooperate to issue e-cashes and at least $(n-t+1)$ out of n administrators do not conspire with the others.
2. Each scheme involves customers, the shops and the n administrators and consists of the following phases: the withdrawal phase, the spending phase and the deposit phase.

3. In the registration phase, customers apply the partially blind threshold signature technique to get their partially blind e-cashes from t honest administrators.
4. In the spending phase, customers generate their real e-cashes from the partially blind e-cashes received in the withdrawal phase and send them to the shops.
5. In the deposit phase, the shops deposit the e-cashes at the bank (any administrator can serve as the bank). The bank will check if the e-cashes were used.

By the previous modifications, the power of a single authority is distributed among several administrators and the issue of e-cashes is controlled by several adniinistrators. The partially blind threshold signature will work when at least t out of n administrators are honest. Since in the withdrawal phase, customers only have to request t members from n administrators, it can meet the real world environments without a single trusted authority or with some absent/dishonest administrators.

The major problem in the on-line e-cash systems proposed by Chaum [1, 6] is the unlimited growth of the bank's database. To cope with the dilemma, our proposed scheme can be directly applied to these schemes to prevent the unlimited growth of the bank's database. Similar to the previous modifications, a customer has to request partially blind threshold signatures as e-cashes from t administrators. When a customer requests partially blind threshold signatures, he negotiates a constant c as the e-coin information with the t signers. For example, $c=(date \cdot amount \cdot expire)$, where *date* is the current date and *amount* is the amount of the e-coin to be withdrawn and *expire* is the expiration date of this coin. At this time, the e-coin becomes (r, s, c). In the deposit phase, the shops deposits the e-coin at the bank. For checking if the e-coin was used, the bank can keep a used e-coin database sorted by the date *date* and check if this e-coin is already in this database. The bank can also record the total amount of e-coin withdrawn in each day. If the total amount of e-coins issued in a particular date was entirely deposited or the e-coins are after the expiration date, the bank can erase the database on this date. By the previous approach, the bank's database can be dramatically reduced.

In real world environments, without the function of partial blindness, the bank has to generate several blind threshold signature schemes and the threshold signatures generated by each scheme will be used as a different amount of e-coins. By our scheme, the bank only has to publish a partially blind threshold signature scheme for all kinds of fixed-amount e-coins. This approach will greatly simplify the e-cash systems since the shadow distribution phase is executed only once and only a set of public parameters has to be published.

5.2. Secure voting schemes

In some small scale elections, different elections may be held several times in one day. In these election schemes [10–12], the tally keys must be regenerated in each voting tally. If the blind threshold signature schemes in [21] are applied to these schemes for distributing the power of the single authority, to regenerate the tally keys is very inefficient since it is time consuming for executing the shadow distribution phase and many public parameters must be published in each tally. To deal with this problem, our proposed scheme can be directly applied to these voting schemes [10–12]. Before a voter requests a partially blind threshold signature as a vote, the signers can first publish a constant c as the current election tag. For example, $c=(date \cdot time)$, where *date* is the date of the election and *time* is the $(time)$th election in this date. When a voter requests partially blind threshold signatures, the constant c will become the negotiated constant for this tally. By this modification, it successfully simplifies the elections, such that, the shadow key distribution phase only has to execute once and the scheme can be used by many elections.

6. Conclusion

We have proposed an efficient partially blind threshold signature scheme based on discrete logarithm. In our scheme, the size of a partially blind threshold signature is the same as that of an individual partially blind signature and the signature verification process is simplified by a group public key. The security of our scheme relies on the hardness 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. Our proposed scheme can be easily applied to current efficient single-authority e-cash systems for distributing the power of a single authority and for preventing the unlimited growth of the bank's database without changing the underlying structure and degrading the overall performance. By our scheme, the bank only has to publish a partially blind threshold signature scheme for all kinds of fixed amount e-cashes, which will greatly simplify the e-cash systems.

References

[1] D. Chaum, Blind signatures for untraceable payments, in: Advances in Cryptology: Proc. Crypt'82, Plenum, New York, 1983, pp. 199–203.

[2] J. Camenisch, J. Pivereau, M. Stadler, Blind signatures based on the discrete logarithm problem, in: Advances in Cryptology: Proc. EuroCrypt'94, LNCS 950, Springer, New York, 1995, pp. 428–432.

[3] D. Chaum, T. Pedersen, Wallet databases with observers, in: Advances in Cryptology: Proc. Crypt'92, LNCS 740, Springer, New York, 1993, pp. 89–105.

[4] C. Fan, C. Lei, A multi-recastable ticket scheme for electronic elections, in: Advances in Cryptology-AisaCrypt'96, LNCS 1163, Springer, New York, 1996, pp. 116–124.

[5] P. Horster, M. Michels, H. Petersen, Meta-message recovery and meta-blind signature schemes based on the discrete logarithm problem and their applications, in: Advances in Cryptology-AisaCrypt'94, LNCS 917, Springer, New York, 1994, pp. 224–237.

[6] D. Chaum, Privacy protected payments: unconditional payer and/or payee untraceability, in: Smartcard 2000, North Holland, Amsterdam, 1988.

[7] D. Chaum, T. Pedersen, Transferred cash grows in size, in: Advances in Cryptology: Proc. EuroCrypt'92, LNCS 658, Springer, New York, 1993, pp. 390–407.

[8] N. Ferguson, Single term off-line coins, in: Advances in Cryptology: Proc. EuroCrypt'93, LNCS 765, Springer, New York, 1993, pp. 318–328.

[9] T. Okamoto, K. Ohta, Universal Electronic cash, in: Advances in Cryptology: Proc. Crypt'91, LNCS 576, Springer, New York, 1992, pp. 324–337.

[10] A. Fujioka, T. Okamoto, K. Ohta, A practical secret voting scheme for large scale elections, Advances in Cryptology: Proc. AusCrypt'92, LNCS 718, Springer, New York, 1992, pp. 244–251.

[11] W. Juang, C. Lei, A secure and practical electronic voting scheme for real world environments, IEICE Trans. Fundamentals E80A (1) (1997) 64–71.

[12] K. Sako, Electronic voting scheme allowing open objection to the tally, IEICE Trans. Fundamentals E77A (1) (1994) 24–30.

[13] R.L. Rivest, A. Shamir, L. Adelman, A method for obtaining digital signatures and public key cryptosystem, Commun. ACM 21 (2) (1978) 120–126.

[14] T. ElGamal, A public key cryptosystem and a signature scheme based on discrete logarithm, IEEE Trans. Inf. Theory IT31 (4) (1985) 469–472.

[15] NIST FIPS PUB XX, Digital signature standard, National Institute of Standards and Technology, US Department of Commerce, DRAFT, 1993.

[16] S. Pohlig, M.E. Hellman, An improved algorithm for computing logarithms over GF(p) and its cryptographic significance, IEEE Trans. Inf. Theory IT24 (1978) 106–110.

[17] D. Chaum, E. van Heyst, Group signatures, in: Advances in Cryptology: Proc. EuroCrypt'91, LNCS 547, Springer, New York, 1991, pp. 257–165.

[18] J. Camenish, M. Stadler, Efficient group signature schemes for large groups, Advances in Cryptology: Proc. Crypt'97, Springer, New York, 1997, pp. 410–424.

[19] J. Camenish, M. Stadler, Efficient and generalized group signatures, Advances in Cryptology: Proc. EuroCrypt'97, Springer, New York, 1997, pp. 465–479.

[20] L. Chen, T.P. Pedersen, New group signature schemes, Advances in Cryptology: Proc. EuroCrypt'94, LNCS 950, Springer, New York, 1995, pp. 171–181.

[21] W. Juang, C. Lei, Blind threshold signatures based on discrete logarithm, in: Proc. Second Asian Computing Science Conference on Programming, Concurrency and Parallelism, Networking and Security, LNCS 1179, Springer, New York, 1996, pp. 172–181.

[22] M. Abe, E. Fujisaki, How to date blind signatures, Advances in Cryptology-AisaCrypt'96, LNCS 11631, Springer, New York, 1996, pp. 244–251.

[23] K. Nyberg, R.A. Rueppel, Message recovery for signature schemes based on the discrete logarithm problem, Advances in Cryptology: Proc. EuroCrypt'94, LNCS 950, Springer, New York, 1995, pp. 182–193.

[24] R.C. Merkle, One way hash functions and DES, Advances in Cryptology: Proc. Crypt'89, LNCS 435, Springer, New York, 1990, pp. 428–446.

[25] NIST FIPS PUB 180, Secure hash standard, National Institute of Standards and Technology, US Department of Commerce, DRAFT, 1993.

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

[27] W. Diffle, M. Hellman, New directions in cryptography, IEEE Trans. Inf. Theory 22 (6) (1976) 644–654.

[28] W. Stallings, Network and Internetwork Security, Prentice-Hall, Englewood Cliffs, NJ, 1995.

Wen-Shenq Juang was born in Taichung, Taiwan in 1969. He received his BSc degree in Computer Science and Information Engineering from Tatung Institute of Technology, Taiwan, in l99l, and MSc degree in Computer Information Science from National Chiao Tung University, Taiwan, in l993. He is now a PhD candidate of electrical engineering at National Taiwan, current research interests include information security and cryptographic protocols in distributed enviromnents. He is also a member of the Chinese Cryptology and Information Security Association.

Chin-Laung Lei was born in Taipei, Taiwan on 9 January 1958. He received a BSc degree in electrical engineering from National Taiwan University in 1980 and a PhD degree in computer science from the Univtrsity of Texas at Austin in 1986. From 1986 to 1988, he was an assistant professor of the computer and information science department at the Ohio State University, Columbus, OH, USA. In 1988, he joined the department of electrical engineering, National Taiwan University, where he is now a professor. His current research interests include network and computer security, parallel and distributed processing, operating system design, and formal semantics of concurrent programs. Dr. Lei is a member of the Institute of Electrical and Electronic Engineers and the Association for Computing Machinery.