1000-9825/2002/13(09)1757-08 ©2002 Journal of Software 软 件 学 报 Vol.13, No.9

A Threshold Undeniable Signature Scheme Without a Trusted Party*

WANG Gui-lin, QING Si-han

(*Engineering Research Center for Information Security Technology, The Chinese Academy of Sciences, Beijing* 100080, *China*);

(*State Key Laboratory of Information Security, Institute of Software, The Chinese Academy of Sciences, Beijing* 100080, *China*)

E-mail: glwang@ercist.iscas.ac.cn; qsihan@yahoo.com

http://www.ercist.ac.cn

Received November 21, 2001; accepted April 18, 2002

Abstract: At Auscrypt'92, Harn and Yang first proposed the conception of (*t*,*n*) threshold undeniable signature, in which only subsets with at least t members can represent a group to generate, confirm or disavow a signature. Later, several schemes are proposed, but none of them is secure. So up to now, how to design a secure (*t*,*n*) threshold undeniable signature scheme is remained an open problem. In this paper, based on discrete logarithm cryptosystem, a secure and efficient (*t*,*n*) threshold undeniable signature scheme without a trusted party is presented. This scheme has an attractive property that member's honesty is verifiable because a publicly verifiable secret sharing scheme is used to distribute secrets and two discrete logarithm equality protocols are used to provide necessary proofs of correctness, which are proposed by Schoenmakers at Crypto'99.

Key words: digital signature; threshold undeniable signature; cryptography; information security

Undeniable signature is a special kind of digital signature with the appealing property that an alleged signature cannot be checked without the cooperation of the signer. (*t*,*n*) *threshold signature* is one kind of group-oriented signature, in which only the subsets with at least *t* members in a group *U* can generate a valid signature and any verifier can simply verify an alleged signature if he/she knows the group public key of *U*. However, in a (*t*,*n*) *threshold undeniable signature* scheme, any subset of *t* members out of *n*, denoted by U_B, can represent the group *U* to generate a signature, but without the cooperation of *t* group members, a verifier cannot verify the validity of an alleged signature even if he knows the group public key. At the same time, any subset of less than *t* members cannot generate, confirm or disavow a signature even if they cooperate maliciously. Generally speaking, a threshold undeniable signature scheme consists of the following three main sub-protocols.

(1) Signing Protocol: *t* members in a subset U_B run this protocol to produce a valid signature for any message, but any attacker *I* cannot forge a valid signature of group *U* with non-negligent possibility unless *I* has corrupted at least *t* members or the group private key has been compromised to *I* (i.e., *nonforgeability*).

(2) Confirmation Protocol: By running this protocol between a subset U_B of *t* members in *U*, i.e. the prover, and a verifier *V*, *V* is convinced that an alleged signature is indeed signed by *U*. Confirmation protocol should satisfy the following three properties.

• Completeness: A valid signature of group *U* will always be accepted by *V* if all the members in U_B and *V* are honest (i.e. they properly act as the protocol described).

---

* Supported by the National Grand Fundamental Research 973 Program of China under Grant No.G1999035810 (国家重点基础研究发展规划 973 项目); the National Natural Science Foundation of China under Grant No.60083007 (国家自然科学基金)

WANG Gui-lin was born in 1968. He is a Ph.D. candidate and an assistant professor at the Institute of Software, CAS. His research interests include cryptography and information security. **QING Si-han** was born in 1939. He is a professor and doctoral supervisor at the Institute of Software, CAS. His research interests are fundamental and application of information security.

• Soundness: Even a cheating subset U_B cannot convince a verifier V to accept a non-valid signature of group U with non-negligent possibility.

• Zero-Knowledge: On input a message and its valid signature, any possible cheating verifier V interacting with a subset U_B does not learn any information aside from the validity of the signature.

(3) Denial Protocol: By running this protocol, prover U_B ensures a verifier V that an alleged signature is not signed by group U. Denial protocol also should satisfy three similar properties as follows.

• Completeness: If all the members in U_B and V are honest, a non-valid signature will always pass through the denial protocol such that V believes that it is not a valid signature of group U.

• Soundness: Even a cheating subset U_B cannot successfully deny a valid signature of U with non-negligent possibility by running denial protocol.

• Zero-Knowledge: On input a message and a non-valid signature, any possible cheating verifier V interacting with a subset U_B does not learn any information aside from the fact that this non-valid signature is in fact not a valid signature of group U.

After the first undeniable scheme was proposed by Chaum and Antwerpen[1], extensive investigations have been studied to this special kind signature. Chaum presented a zero-knowledge undeniable signature scheme with many useful applications[2]. By incorporating both concepts of the undeniable signature and group-oriented signature[3,4], Harn and Yang[5] proposed the conception of (t,n) threshold undeniable signature and designed two concrete schemes in respect of t=1 and t=n. But Langford[6] pointed out that their (n,n) threshold undeniable signature scheme only possesses the security level of $(2,n)$, because any two adjacent members can generate a valid signature. Later, Lin *et al.* presented a general threshold undeniable signature scheme without a trusted party[7], but their scheme is also subjected to the same attack. In 1999, Ref.[8] generalized Chaum's zero-knowledge undeniable signature[2] to a (t,n) threshold undeniable signature scheme, but this scheme has two shortcomings: (a) it needs the help of a trusted party; (b) invalid partial signatures cannot be detected. All these threshold undeniable schemes are based on discrete logarithm cryptosystems, but none of them is secure and does not need the help of a trusted party. So up to now, the problem of designing a secure (t,n) threshold undeniable signature scheme without a trusted party is remained open.

Based on the first undeniable RSA signature scheme[9] and a revised version of Shoup's practical threshold RSA signature scheme[10], Ref.[11] presented the first threshold undeniable RSA signature scheme with a trusted party.

In this paper, based on discrete logarithm cryptosystem, we present a secure and efficient (t,n) threshold undeniable signature scheme without a trusted party. Essentially speaking, our scheme is a generalization of the Chaum and Antwerpen's undeniable scheme[1] to threshold environment. By making use of a publicly verifiable secret sharing (PVSS) scheme, proposed by Schoenmakers[12], and two non-interactive discrete logarithm equality protocols, our scheme has an attractive property that each member's honesty is verifiable in all the following stages: distributing secrets, establishing group public key, generating signature, confirming and disavowing an alleged threshold undeniable signature. We call these two non-interactive discrete logarithm equality protocols as DLE protocol, proposed by Perderson and Chaum[13,14], and DDLE protocol, which is a modified version to a protocol proposed by Stadler[15].

This paper is organized as follows. Several notations are introduced in Section 1. Then, in Section 2, DLE and DDLE protocols are reviewed concisely. Afterwards, the new threshold undeniable signature scheme is described in Section 3. Finally, some brief discussions to our new scheme are given in Section 4.

1 Notations

n members U_i (i=1,2,…,n) consists of a group U and t is the threshold value. Let B denote a subset of size t in the index set {1,2,…,n} and $U_B=\{U_i|i\in B\}$ be a subset of size t in U. The notation $x\in_R X$ means that an element x is selected randomly and uniformly from the set X.

q, p' and p are three primes such that $q|p'-1$ and $p|p'-1$.G_q is the unique multiplicative subgroup of order q in finite field $Z_{p'}$, and $G_{p'}$ is the unique multiplicative subgroup of order p' in finite field Z_p.

H_1, H_2, and H_3 are three hash functions such that $H_2:\{0,1\}^*\to\{0,1\}^l$ and $H_3: Z\to G_{p'}\subseteq Z_p$. Where, l is a security parameter ($l\approx 100$). Then, for every original message M such that $m=H_3(M)\neq 1$, m is a generator of group $G_{p'}$. Such special kind of hash function H_3 can be constructed as follows: after choosing a hash function $H': Z\to Z_{p'}$ and a generator g of $G_{p'}$, we define H_3 as $m=H_3(M)=g^{H'(M)} \bmod p$, $\forall M\in Z$.

2 Discrete Logarithm Equality Protocols

Knowledge proving protocols, especially of which based on the discrete logarithm problems, are extensively used in modern cryptography[16]. In this section, we will describe DLE and DDLE protocol briefly.

2.1 DLE ($g_1,h_1;g_2,h_2;\alpha$) protocol

g_1,g_2,h_1 and h_2 are four public numbers such that g_1,g_2 are two generators of group G_q. The prover P knows a secret number $\alpha\in Z_q^*$ such that $\log_{g_1} h_1=\log_{g_2} h_2=\alpha$, i.e. $h_1=g_1{}^{\alpha} \bmod p'$ and $h_2=g_2{}^{\alpha} \bmod p'$. By running the following DLE($g_1,h_1;g_2,h_2;\alpha$) protocol, the prover P produces necessary proof to convince a verifier V that he indeed knows the secret α but does not reveal which is the α.

(1) P randomly selects $w\in_R Z_q$, computes $a_1=g_1{}^w \bmod p'$, $a_2=g_2{}^w \bmod p'$, $c=H_1(a_1\|a_2)$ and $r=w-\alpha c \bmod q$. P publishes $\text{Proof}_P=(r,c)$ as the proof of knowing the secret α.

(2) V determines whether P knows the secret α by checking $c\equiv H_1(g_1{}^r h_1{}^c\|g_2{}^r h_2{}^c)$.

The completeness of these protocols is obvious, and the soundness and zero-knowledge are consulted to Refs.[13, 14].

2.2 DDLE ($h_1,A;h_2,g,B;\alpha$) protocol

Stadler[15] designed a knowledge protocol to prove that a discrete logarithm is equal to a double discrete logarithm. In this subsection, we present an improved version of Stadler's protocol and call it as DDLE protocol. This protocol is constructed under the same frame of Stadler's, but it reveals less information. Therefore, it is at least as secure as Stadler's original protocol. In addition, the structural format of proof is also different with Stadler's.

Let h_1, h_2 be two public generators of G_q (i.e. two elements of order q in $Z_{p'}$). Suppose that at most the prover P knows the discrete logarithm $\log_{h2} h_1$. g is a public generator of $G_{p'}$ (i.e. an element of order p' in Z_p) such that computing discrete logarithms to base g is difficult.

Now, suppose that the prover P knows a secret $\alpha\in Z_q$ such that two public numbers A and B satisfy $A=h_1{}^{\alpha} \bmod p'$ and $B=g^{h_2{}^{\alpha}} \bmod p$. Then P can run the following DDLE($h_1,A;h_2,g,B;\alpha$) protocol to convince a verifier V that he indeed knows such α but does not reveal which is the α.

(1) P first selects l random numbers $w_i\in_R Z_q$ and computes the following $2l$ values:

$$a_{1i}=h_1^{w_i} \bmod p',\quad a_{2i}=g^{h_2^{w_i}} \bmod p,\quad i=1,2,\ldots,l.$$

(2) Then P evaluates the following hash function value c as the challenge:

$$\underline{c}=H_2(A\|B\|a_{11}\|a_{21}\|\ldots\|a_{1i}\|a_{2i}\|\ldots\|a_{1l}\|a_{2l}) \tag{1}$$

(3) P computes l responses: $r_i=w_i-c_i\,\alpha \bmod q$ (i=1,2,…l) where c_i is the i-th bit of c.

(4) At last, P publishes the proof $\text{Proof}_p=(c,r_1,r_2,\ldots,r_l)$

(5) When V want to check whether P knows the secret α he first computes a_{1i} and a_{2i} by using of Proof_P:

$$a_{1i}=h_1{}^{r_i}A^{c_i} \bmod p',\qquad a_{2i}=(g^{1-c_i}B^{c_i})^{(h_2^{r_i})} \bmod p. \tag{2}$$

Then, V checks whether equation (1) holds. If yes, he receives the knowledge proving of prover P; otherwise rejects it.

Theorem 1. (Completeness of DDLE protocol) If the prover P and the verifier V all are honest, then V always receives P's knowledge proving.

Proof. Because P is honest, so he selects l random numbers $w_i \in_R Z_q$ such that

$$a_{1i} = h_1^{w_i} \bmod p', \quad a_{2i} = g^{h_2^{w_i}} \bmod p. \tag{3}$$

Then, P computes the challenge c by Eq.(1) and the l responses r_i by $r_i = w_i - c_i\alpha \bmod q$. On the other hand, in the above step (5), the verifier V computes a_{1i} and a_{2i} from Eq.(2) by using the proof $(c,r_1,r_2,\ldots,r_l)$. Note that the following equations hold:

$$a_{1i} = h_1^{r_i} A^{c_i} \bmod p' = h_1^{w_i - c_i\alpha}(h_1^{\alpha})^{c_i} \bmod p' = h_1^{w_i} \bmod p';$$

$$a_{2i} = (g^{1-c_i} B^{c_i})^{(h_2^{r_i})} \bmod p = g^{(c_i h_2^{\alpha}+1-c_i)h_2^{w_i - c_i\alpha}} \bmod p = g^{h_2^{w_i}} \bmod p, \text{ whether } c_i = 0 \text{ or } c_i = 1.$$

It is known that V obtains the same values a_{1i} and a_{2i} as P does (i.e. equation (3)). Therefore, V founds that equation (1) holds, i.e. V always receives the knowledge proving of P if they run the DDLE protocol honestly.

About the soundness and zero-knowledge of DDLE protocol, similar discussions can be addressed as Ref.[15] did.

3 Description of the Proposed Scheme

In this section, we present a threshold undeniable signature scheme based on discrete logarithm cryptosystem without any trusted party. In the design of this scheme, we adopt the publicly verifiable secret sharing scheme (simple denoted by PVSS), proposed by Schoenmakers in Ref.[12], to make our scheme satisfying the attractive property that the honesty of each member is verifiable. More specially speaking, we use the DLE and DDLE protocols described in last section to construct necessary proofs such that the operations of each member in all the following phases are verifiable: group public key generation, secret distribution, threshold undeniable signature generation, confirmation and denial.

Stage 1. System initialization

Group U selects the system public parameters $p, p', q, g, \alpha, \beta, H_3$ such that:

(1-1) p, p', and q are large primes such that $q \mid p'-1$ and $p' \mid p-1$.

(1-2) g is a generator of order p' in finite field Z_p.

(1-3) α and β are two generators of order q in finite field $Z_{p'}$ and nobody knows the discrete logarithm $\log_\alpha \beta$ and $\log_\beta \alpha$. As Gennaro et al. pointed out in the Section 4.1 of Ref.[17], a generic distributed coin flipping protocol will accomplish the generating of α and β.

(1-4) H_3 is a hash function from Z to $G_{p'}$ (as described in Section 1).

Stage 2. Secrets distribution

(2-1) Each member U_i selects his private key $x_i \in_R Z_q^*$, then computes and registers the following t_i as his public key (t_i is a generator of G_q):

$$t_i = \alpha^{x_i} \bmod p'. \tag{4}$$

(2-2) Member U_i randomly chooses a polynomial $f_i(x)$ with order at most t–1: $f_i(x) = \sum_{j=0}^{t-1} a_{ij} x^j \in Z_q[x]$, where $a_{ij} \in_R Z_q$.

(2-3) Member U_i computes y_i and Y_i as follows:

$$y_i = \alpha^{f_i(0)} \bmod p', \qquad Y_i = g^{y_i} \bmod p.$$

U_i signs Y_i and publishes it, but keeps $f_i(0)$, i.e., a_{i0} and y_i secretly.

(2-4) U_i runs the PVSS protocol[12] to distribute the secret y_i. Following is the procedure in detail.

In the first, U_i publishes C_{ij} as his commitment to each coefficient of the polynomial $f_i(x)$ and T_{ik} as the encrypted shadow sub-key for member U_k:

$$C_{ij}=\beta^{a_{ij}} \bmod p',\quad \forall j\in\{0,1,...,t-1\};\qquad T_{ik}=t_k^{f_i(k)} \bmod p',\quad \forall i\in\{1,2,...,n\}. \tag{5}$$

Now, let

$$X_{ik}=\prod_{j=0}^{t-1}C_{ij}^{k^j} \bmod p'\ (=\beta^{\sum_{j=0}^{t-1}a_{ij}\cdot k^j} \bmod p'=\beta^{f_i(k)} \bmod p'). \tag{6}$$

It is easy to see that every member can work out the values of X_{ik} by using the public information C_{ij} ($0\le j\le t-1$).

Then, U_i shows that all the encrypted shadow sub-keys T_{ik} ($1\le k\le n$) are consistent by constructing a proof of knowledge of the unique $f_i(k)\,(1\le k\le n)$ satisfying:

$$\log_\beta X_{ik}=\log_{t_k}T_{ik}(=f_i(k)),\quad k=1,2,...,n.$$

For this seek, applying Fiat-Shamir's technique[18], U_i selects n random numbers $w_{ik}\in_R Z_q$ to compute the following values a_{ik} and $\bar{a}_{ik}$:

$$a_{ik}=\beta^{w_{ik}} \bmod p',\qquad \bar{a}_{ik}=t_k^{w_{ik}} \bmod p',\qquad k=1,2,...,n.$$

And then U_i compute the challenge c_i as follows:

$$c_i=H_1(X_{i1}\|...\|X_{in}\|T_{i1}\|...\|T_{in}\|a_{i1}\|...\|a_{in}\|\bar{a}_{i1}\|...\|\bar{a}_{in}). \tag{7}$$

Using the challenge c_i, U_i computes the response r_{ik} for member U_k:

$$r_{ik}=w_{ik}-f_i(k)c_i \bmod q,\qquad k=1,2,...,n.$$

Finally, U_i constructs the knowledge proof as following:

$$\text{Proof}_i=(c_i,r_{i1},r_{i2},...,r_{in}).$$

Each member can verify whether U_i distributes secret honestly by checking the equality (7). Here is the reason. By using the public information $C_{ij},X_{ik},t_k,T_{ik},r_{ik}$ and c_i, he can work out the values a_{ik} and $\bar{a}_{ik}$ as follows:

$$a_{ik}=\beta^{r_{ik}}X_{ik}^{c_i} \bmod p',\qquad \bar{a}_{ik}=t_k^{r_{ik}}T_{ik}^{c_i} \bmod p',\qquad k=1,2,...,n.$$

(2-5) Now, U_i sends $f_i(k)$ to U_k secretly. U_k checks whether the following equality holds:

$$\alpha^{f_i(k)}\equiv T_{ik}^{1/x_k} \bmod p'.$$

(2-6) If any member fails in above steps, then the total scheme aborts. Otherwise, all members in group U pass through above steps without any dissent, then U_i computes values $f(i),X_i,T_i$ and S_i as follows:

$$\begin{aligned}
&f(i)=\textstyle\sum_{k=1}^n f_k(i) \bmod q;\\
&X_i=\textstyle\prod_{k=1}^n X_{ki} \bmod p'\ (=\beta^{\sum_{k=1}^n f_k(i)} \bmod p'=\beta^{f(i)} \bmod p');\\
&T_i=\textstyle\prod_{k=1}^n T_{ki} \bmod p'\ (=t_i^{\sum_{k=1}^n f_k(i)} \bmod p'=t_i^{f(i)} \bmod p');\\
&S_i=T_i^{1/x_i} \bmod p'\ (=(t_i^{f(i)})^{1/x_i} \bmod p'=\alpha^{f(i)} \bmod p').
\end{aligned} \tag{8}$$

Here, S_i is the sub-key that U_i gets. In addition, member U_i publishes T_i and X_i publicly, but keeps S_i and $f(i)$ secretly.

Stage 3. Generation of the group public key

(3-1) Using the public information Y_k ($k=1,2,...,n$), all n members in group U connect in a ring and run the following RING1 protocol to generate the group public key Y as

$$Y=g^{y_1y_2\cdots y_n} \bmod p=g^y \bmod p. \tag{9}$$

where $y=y_1y_2...y_n \bmod p'$ is the group private key and nobody knows it.

RING1 (g,y_i,Y_i,Y) **Protocol**

For convenience, we assume these n members connect in the following order:

$$U_1\Rightarrow U_2\Rightarrow\cdots U_{i-1}\Rightarrow U_i\Rightarrow U_{i+1}\cdots\Rightarrow U_n\Rightarrow U_1.$$

Step 1. U_1 uses Y_1 to sign a public message m_0 agreed by group U as below (m_0 can be selected as the identity of group U or member U_1, or anything else).

$$Y_0=m_0^{y_1} \bmod p.$$

And let $\overline{Y}_0 = g, \overline{Y}_1 = \overline{Y}_0^{y_1} \bmod p\ (= Y_1 = g^{y_1} \bmod p)$. Now, U_1 runs DLE $(m_0, Y_0; \overline{Y}_0, \overline{Y}_1; y_1)$ protocol and broadcasts $(Y_0, \overline{Y}_1, \text{Proof}_{U_1})$.

Step 2. By using $(Y_0, \overline{Y}_1, \text{Proof}_{U_1})$, U_2 (and each member) checks whether $\log_{m_0} Y_0 = \log_{\overline{Y}_0} \overline{Y}_1\ (= y_1)$. If not, he declares this fact and stops running the protocol. Otherwise, U_2 first computes

$$\overline{Y}_2 = \overline{Y}_1^{y_2} \bmod p.$$

Then he runs DLE $(g, Y_2; \overline{Y}_1, \overline{Y}_2; y_2)$ protocol, constructs proof and broadcasts $(\overline{Y}_2, \text{Proof}_{U_2})$.

Step i ($3 \le i \le n$). By using $(\overline{Y}_{i-1}, \text{Proof}_{U_{i-1}})$, U_i (and each member) checks whether $\log_g Y_{i-1} = \log_{\overline{Y}_{i-2}} \overline{Y}_{i-1}\ (= y_{i-1})$. If not, he declares this fact and stops running the protocol. Otherwise, U_i first computes

$$\overline{Y}_i = \overline{Y}_{i-1}^{y_i} \bmod p.$$

Then he runs DLE $(g, Y_i; \overline{Y}_{i-1}, \overline{Y}_i; y_i)$ protocol, constructs proof and broadcasts $(\overline{Y}_i, \text{Proof}_{U_i})$.

Step n+1. By using $(\overline{Y}_n, \text{Proof}_{U_n})$, all members check whether $\log_g Y_n = \log_{\overline{Y}_{n-1}} \overline{Y}_n\ (= y_n)$. If yes, this protocol outputs the following Y as the group public key:

$$Y = \overline{Y}_n = g^{y_1 y_2 \cdots y_n} \bmod p = g^y \bmod p.$$

(3-2) After the generation of Y, $(p, p', q, g, \alpha, \beta, H_1, H_2, H_3, Y, ID_U, t_i)$ can be submitted to a Certificate Authority for getting a registered certificate of the group public key of group U.

Stage 4. Generation of threshold undeniable signature

If t members in U_B want to sign message m, then each U_i ($i \in B$) does as follows.

(4-1) Each U_i ($i \in B$) first computes

$$S'_{Bi} = S_i^{C_{Bi}} \bmod p'\ (= \alpha^{\bar{f}(i)} \bmod p'). \tag{10}$$

where C_{Bi} and $\bar{f}(i)$ are defined respectively by

$$C_{Bi} = \prod_{j \in B \setminus \{i\}} \frac{j}{j-i} \bmod q, \quad \text{and} \quad \bar{f}(i) = C_{Bi} \cdot f(i) \bmod q.$$

(4-2) All these t members U_i ($i \in B$) connect in a ring and run the following RING2 protocol to generate threshold undeniable signature z. For convenience, we assume that they are the first t members in group U (i.e. $B=\{1,2,\ldots,t\}$) and connect in the following order:

$$U_1 \Rightarrow U_2 \Rightarrow \cdots U_{i-1} \Rightarrow U_i \Rightarrow U_{i+1} \cdots \Rightarrow U_t \Rightarrow U_1.$$

RING2 $(m; t_i, T_i^{C_{Bi}}; \alpha, z_{i-1}, z_i; z)$ **Protocol**

Step 1. U_1 computes his partial signature z_1 as follows:

$$z_1 = m^{S'_{B1}} \bmod p\ (= m^{\alpha^{\bar{f}(1)}} \bmod p).$$

Then he runs DDLE $(t_1, T_1^{C_{B1}}; \alpha, m, z_1; \bar{f}(1))$ protocol and broadcasts $(\text{Proof}_{U_1}, z_1)$. Each member can verify whether $\log_{t_1}(T_1^{C_{B1}}) = \log_\alpha(\log_m z_1)\ (= \bar{f}(1))$.

Step $i\ (2 \le i \le t)$. When U_i sees $(\text{Proof}_{U_{i-1}}, z_{i-1})$, according to Eqs.(2) and (1), he verify whether member U_{i-1} generated z_{i-1} properly. If not, U_i declares this fact and stops running the protocol. Otherwise, U_i computes his partial signature z_i:

$$z_i = z_{i-1}^{S'_{Bi}} \bmod p\ (= z_{i-1}^{\alpha^{\bar{f}(i)}} \bmod p). \tag{11}$$

Then he runs DDLE $(t_i, T_i^{C_{Bi}}; \alpha, z_{i-1}, z_i; \bar{f}(i))$ protocol and broadcasts $(\text{Proof}_{U_i}, z_i)$. Each member can verify whether $\log_{t_i}(T_i^{C_{Bi}}) = \log_\alpha(\log_{z_{i-1}} z_i)\ (= \bar{f}(i))$.

(4-3) If z_t is generated properly, then we define $z = z_t$ as the threshold undeniable signature of group U on message m. From above description, it is not difficult to see the following equation holds:

$$z = z_t = m^{\prod_{i=1}^{t} S'_{Bi}} \bmod p = m^y \bmod p. \tag{12}$$

Stage 5. Confirmation of threshold undeniable signature

It is the goal of above RING2 $(m;t_i,T_i^{C_{Bi}};\alpha,z_{i-1},z_i;z)$ protocol that t members $U_i\ (i\in B)$ generate the threshold undeniable signature z defined by equation (12). At the same time, each member runs the DDLE $(t_i,T_i^{C_{Bi}};\alpha,z_{i-1},z_i;\bar{f}(i))$ protocol to generate necessary proof such that U_i's neighbor U_{i+1} (and all other members) can verify the validity of partial signature z_i. In our confirmation protocol, t members $U_i\ (i\in B$ and $|B|=t)$ of U need to compute the following value R as the response to a challenge W provided by the verifier V:

$$R=W^{\prod_{i\in B}(S'_{Bi})^{-1}} \bmod p\ \ (=W^{y^{-1}} \bmod p).$$

Note that every member can compute $T_i^{-C_{Bi}}$ and that we have the following two equations

$$\begin{aligned}(S'_{Bi})^{-1}&=S_i^{-C_{Bi}} \bmod p'=\alpha^{-\bar{f}(i)} \bmod p';\\ T_i^{-C_{Bi}}&=t_i^{-C_{Bi}\cdot f(i)} \bmod p'=t_i^{-\bar{f}(i)} \bmod p'.\end{aligned} \tag{13}$$

It is easy to know that these t members can compute the response R by running RING2 $(W;t_i,T_i^{-C_{Bi}};\alpha,R_{i-1},R_i;R)$ protocol. Where, R_i is defined by

$$R_i=R_{i-1}^{(S'_{Bi})^{-1}} \bmod p\ \ (=(R_{i-1})^{\alpha^{-\bar{f}(i)}} \bmod p),\quad \text{and}\quad R_0=W.$$

Now, we present the confirmation protocol as below.

(5-1) Verifier V selects two random number $a,b\in_R Z_{p'}$, and sends the following W to all the t members $U_i\ (i\in B)$:

$$W=z^aY^b \bmod p. \tag{14}$$

where (m, z) is an alleged signature message pair and Y is the group public key of U.

(5-2) t members $U_i\ (i\in B)$ connect in a ring to run RING2 $(W;t_i,T_i^{-C_{Bi}};\alpha,R_{i-1},R_i;R)$ protocol. If success, they send the output R to V.

(5-3) V accepts the signature (m, z) if and only if the following equality holds:

$$R\equiv m^ag^b \bmod p. \tag{15}$$

Stage 6. Denial of threshold undeniable signature

If verification Eq.(15) does not hold after V and t members have run the confirmation protocol, then they run the following denial protocol to convince V that signature z is not signed by group U. Like the denial protocol in Ref.[1], two successful denials to an alleged signature (m, z) serves as the denial protocol.

(6-1) By running the confirmation protocol with t members of group U for two times, V gets two triples (R, a, b) and $(\bar{R},\bar{a},\bar{b})$, but any of them does not satisfy the verification Eq.(15). Then, verifier V believes that (m, z) is in fact not a signature of group U if and only if the following equality holds:

$$(Rg^{-b})^{\bar{a}}\equiv(\bar{R}g^{-\bar{b}})^a \bmod p. \tag{16}$$

4 Analysis of the Proposed Scheme

Now we briefly discuss the validity and security of our threshold undeniable signature scheme. In the first, it is easy to know that our scheme is correct, i.e. if t honest members generate valid partial signatures, then the getting undeniable signature will be passed through the confirmation protocol. In the second, we combine the Shamir's secret sharing scheme[19] and Schoenmakers' PVSS[12] together to distribute secrets such that less than t members cannot deduce the group private key and each member has to distribute secrets honestly otherwise his cheating behavior will be detected. In the third, the group public key can be generated efficiently and securely by running RING1 protocol because DLE protocol are employed to provide proof of correctness. In the last, each member has to run DDLE protocol to produce necessary proof in all the following stages: generation, confirmation and denial of a threshold undeniable signature. Any cheater in these stages will be detected.

Therefore, based on discrete logarithm cryptosystem, we have proposed a valid and secure threshold undeniable signature scheme without a trusted party.

References:

[1] Chaum, D., van Antwerpen, H. Undeniable signatures. In: Brassard, G., ed. Proceedings of the Advances in Cryptology-Crypto’89. LNCS 435, Berlin: Springer-Verlag, 1989. 212~216.

[2] Chaum, D. Zero-Knowledge undeniable signatures. In: Damgard, I.B., ed. Proceedings of the Advances in Cryptology-Eurocrypt’90. LNCS 473, Berlin: Springer-Verlag, 1991. 458~464.

[3] Desmedt, Y. Society and group oriented cryptography: a new concept. In: Pomerance, C., ed. Proceedings of the Advances in Cryptology-Crypto’87. LNCS 293, Berlin: Springer-Verlag, 1988. 120~127.

[4] Desmedt, Y., Frankel, Y. Threshold cryptosystems. In: Brassard, G., ed., Proceedings of the Advances in Cryptology-Crypto’89. LNCS 435, Berlin: Springer-Verlag, 1990. 307~315.

[5] Harn, L., Yang, S. Group-Oriented undeniable signature schemes without the assistance of a mutually trusted party. In: Seberry, J., Zheng, Y., eds. Proceedings of the Advances in Cryptology-Auscrypt’92. LNCS 718, Berlin: Springer-Verlag, 1993. 133~142.

[6] Langford, S.K. Weakness in some threshold cryptosystems. In: Koblitz, N., ed. Proceedings of the Advances in Cryptology-Crypto’96. LNCS 1109. Berlin: Springer-Verlag, 1996. 74~82.

[7] Lin, C.-H., Wang, C.-T., Chang, C.-C. A group-oriented (t,n) undeniable signature scheme without trusted center. In: Pieprzyk, J., Seberry, J., eds. Proceedings of the Information Security and Privacy, ACISP’96. LNCS 1172, Berlin: Springer-Verlag, 1996. 266~274.

[8] Lee, N.-Y., Hwang, T. Group-Oriented undeniable signature schemes with a trusted center. Computer Communications, 1999,22: 730~734.

[9] Gennaro, R., Krawczyk, H., Rabin, T. RSA-Based undeniable signature. In: Kaliski, B., ed. Proceedings of the Advances in Cryptology-Crypto’97. LNCS 1294, Berlin: Springer-Verlag, 1997. 132~148.

[10] Shoup, V. Practical threshold signatures. In: Vaudenay, S., ed., Proceedings of the Advances in Cryptology–Eurocrypt’2000. LNCS 1807, Berlin: Springer-Verlag, 2000. 207~220. http://www.shoup.net/papers/

[11] Wang, Gui-lin, Qing, Si-han, Wang, Ming-sheng, *et al*. Threshold undeniable RSA signature scheme. In: Qing, S., Okamoto, T., Zhou, J. eds. Proceedings of the Information and Communications Security (ICICS 2001). LNCS 2229, Berlin: Springer-Verlag, 2001. 221~232.

[12] Schoenmakers, B. A simple publicly verifiable secret sharing scheme and its application to electronic voting. In: Wiener, M. ed. Proceedings of the Advances in Cryptology-Crypto’99. LNCS 1666, Berlin: Springer-Verlag, 1999. 148~164.

[13] Chaum, D., Pedersen, T.P. Transferred cash grows in size. In: Rueppel, R.A., ed. Proceedings of the Advances in Cryptology-Eurocrypt’92. LNCS 658, Berlin: Springer-Verlag, 1993. 390~407.

[14] Chaum, D., Pedersen, T.P. Wallet databases with observers. In: Brickell, E.F., ed. Proceedings of the Advances in Cryptology-Crypto’92. LNCS 740, Berlin: Springer-Verlag, 1993. 89~105.

[15] Stadler, M. Publicly verifiable secret sharing. In: Maurer, U.M. ed. Proceedings of the Advances in Cryptology-Eurocrypt’96. LNCS 1070, Berlin: Springer-Verlag, 1996. 191~199.

[16] Camenisch, J., Michels, M. Proving in zero-knowledge that a number is the product of two safe primes. In: Stern, J., ed. Proceedings of the Advances in Cryptology-Eurocrypt’99. LNCS 1592, Berlin: Springer-Verlag, 1999. 107~122.

[17] Gennaro, R., Jarecki, S., Krawczyk, H., *et al*. Secure distributed key generation for discrete-log based cryptosystems. In: Stern, J., ed. Proceedings of the Advances in Cryptology-Eurocrypt’99. LNCS 1592, Berlin: Springer-Verlag, 1999. 295~310.

[18] Fiat, A., Shamir, A. How to prove yourself: practical solutions to indentification and signature problems. In: Odlyzko, A.M., ed. Proceedings of the Advances in Cryptology-Crypto’86. LNCS 263, Berlin: Springer-Verlag, 1987. 186~194.

[19] Shamir, A. How to share a secret. Communications of the ACM, 1979,22(11):612~613.

不需要可信任方的门限不可否认签名方案

王贵林， 卿斯汉

(中国科学院 信息安全技术工程研究中心,北京 100080);

(中国科学院 软件研究所 信息安全国家重点实验室,北京 100080)

摘要: 在 1992 年澳大利亚密码会议上, Harn and Yang 第一次提出了(t,n)门限不可否认签名的概念.其中,只有成员个数不少于 t 的子集才能代表群体产生、确认和否认签名.随后,一些研究者又提出了几个方案,但这些方案都是不安全的.因此,到目前为止,怎样设计一个安全的(t,n)门限不可否认签名方案仍然是个公开问题.提出了一个基于离散对数密码系统的(t,n)门限不可否认签名方案.该方案不仅安全、高效,而且不需要可信任方.另外,方案还具有一个很好的性质,即成员的诚实性是可以验证的.这是由于在分发密钥时,采用了 Schoenmakers 在 1999 年美洲密码会议上提出的可公开验证秘密共享方案和两个用来提供正确性证据的离散对数恒等式协议.

关键词: 数字签名;门限不可否认签名;密码学;信息安全

中图法分类号: TP309 文献标识码: A