Threshold Digital Signatures based on Discrete Logarithm

Choonsik Park

Abstract

This paper presents a (*k*,*n*) threshold digital signature scheme with no trusted dealer. Our idea is to use the ElGamal signature scheme modified for group use. Among many digital signature schemes, our modification has a nice property for our purpose. We also show a (*k*,*n*) threshold fail stop signature scheme and two (*k*,*n*) threshold undeniable signature schemes. We use [10] as the original fail stop signature scheme, and use [3] and [2] as the original undeniable signature schemes. Since all these schemes are based on the discrete log problem, we can use the same technique.

I. Introduction

The notion of group oriented cryptosystems was introduced by [4]. In a (*k*,*n*) threshold public key cryptosystem, the receiver is a group of *n* members such that *k* out of *n* members must cooperate for decryption keeping the decryption key secret. [5] showed a (*k*,*n*) threshold ElGamal cryptosystem. In this system, a trusted dealer was necessary. [13] showed such a public key cryptosystem with no trusted dealer. [6] showed that a (*k*,*n*) threshold digital signature scheme for RSA could only exist with a trusted dealer.

On the other hand, undeniable signature schemes were introduced by [3]. Unlike digital signature schemes, an undeniable signature scheme consists of a signature issuing phase and a signature verification phase. In the signature issuing phase, a signer issues an undeniable signature. However, the signature cannot be verified without the help of the signer. In the signature verification phase, a protocol is executed between the signer and a verifier. The validity of the undeniable signature is verified by this protocol. Group oriented undeniable signature schemes were studied by [12] and [11]. [12] showed how the signer can distribute a part of his secret key to *n* agents such that any *k* of these can verify a signature. That is, only the signature verification phase is group oriented. [11] showed an (*n*,*n*) threshold undeniable signature scheme such that both the signature issuing phase and the signature verification phase are group oriented. However, all *n* members must cooperate. How to realize a fully (*k*,*n*) threshold undeniable signature scheme is an open problem [11].

(*k*,*n*) threshold signature schemes with no trusted dealer are not known so far for both digital signature schemes and undeniable signature sche mes.

This paper presents (*k*,*n*) threshold signature schemes with no trusted dealer. We first show a modification of ElGamal signature scheme to make it suitable for group use. Our idea is to use the modified ElGamal signature scheme as the original digital signature scheme. Among many digital signature schemes, our modification has a nice property for our purpose. We use this modified scheme as the digital signature scheme. We also show a (*k*,*n*) threshold fail stop signature scheme and two (*k*,*n*) threshold undeniable signature schemes. We use [10] as the original fail stop signature scheme, and use [3] and [2] as the original undeniable signature schemes. Since all these schemes are based on the discrete log problem, we can use the same technique.

In a fail stop signature, the signer can show a proof of the forgery if someone makes a forgery. In an undeniable signature scheme, the signer's cooperation is necessary to verify the validity or the invalidity of a signature. This scheme is useful to prevent illegal copy of software, for example.

Group digital signature schemes are important, for example, in the system of Kohnfelder's public key certificate. Let E_A be the public key of Alice. In that system, a trusted center gives the digital signature *Sign*(Alice, E_A) to Alice. If a dishonest center gives *Sign*(Bob, E_A) to Alice, Alice could

Manuscript received July 21, 1995; accepted May 17, 1996.

The author is with Electronics and Telecommunications Research Institute, Taejon, Korea.

impersonate Bob to anyone. Thus, we should distribute the power of the signer.

II. Preliminaries

1. Fail Stop Signature Scheme [10]

Let g_1 and g_2 be primitive elements of *GF(p)*.

(secret key) $s_1, s_2, s_3, s_4 (0 < s_i < p-1)$

(public key) $y_1 (= g_1^{s_1} g_2^{s_2})$, $y_2 (= g_1^{s_3} g_2^{s_4})$

(plaintext) $m \in Z_{p-1}$

(signature) (z_1, z_2), where

$$z_1 = s_1 + ms_3 \mod p-1$$

$$z_2 = s_2 + ms_4 \mod p-1$$

(verification) $y_1 y_2^m = g_1^{z_1} g_2^{z_2} \mod p$

(proof of forgery)

If an opponent forges a signature ($\hat{z}_1$, $\hat{z}_2$), the signer shows (z_1, z_2). Even if the opponent is infinitely powerful, $(z_1, z_2) \neq (\hat{z}_1, \hat{z}_2)$ with overwhelming probability.

2. Undeniable Signature Scheme [3]

We use *GF(p)* such that *p*-1 is divided by a large prime *q*. Let G_q be the subgroup of $GF(p)^*$ of order *q* and let *g* be a generator of G_q.

(secret key) $s \ (0 < s < q-1)$

(public key) $y \ (= g^s)$

(plaintext) $m \in G_q$

(undeniable signature) $z \ (= m^s)$

(confirmation protocol)

Suppose that *z* is valid. The signer P proves this fact to a verifier *a* by executing the following confirmation protocol.

(step 1) V randomly selects two integers *a* and *b* from Z_q, and computes

$$w = z^a y^b.$$

w is sent to P.

(step 2) P computes

$$R = w^{s^{-1}}$$

where s^{-1} is the multiplicative inverse of *s* mod *q*. *R* is sent back to V.

(step 3) V checks whether

$$R = m^a g^b. \qquad (1)$$

If this equation is satisfied, the signature is valid.

(disavowal protocol)

Suppose that *z* is invalid. P proves this invalidity by the following protocol.

(step 1) V randomly selects two integers *a* and *b* from Z_q, and computes

$$w = z^a y^b.$$

w is sent to P.

(step 2) P computes

$$R = w^{s^{-1}}.$$

R is sent back to V.

Eq.(1) is not satisfied in this case because *z* is assumed to be invalid.

(step 3) V randomly selects *c* and *d*, and calculates

$$w' = z^c y^d.$$

w' is sent to P.

(step 4) P computes

$$R' = w'^{s^{-1}}.$$

R' is sent back to V.

(step 5) V checks whether

$$(Rg^{-b})^c = (R'g^{-d})^a.$$

The equality means that P is answering consistently and the signature *z* is invalid.

3. ZK Undeniable Signature Scheme [2]

We use GF(*p*) such that p-1 is divided by a large prime *q*. Let G_q be the subgroup of GF(*p*)* of order *q* and let *g* be a generator of G_q. Let *g* be a primitive element of *GF(p)*.

(secret key) $s \ (0 < s < q-1)$

(public key) $y \ (= g^s)$

(plaintext) $m \in G_q$

(undeniable signature) $z \ (= m^s)$

(confirmation protocol)

(step 1) A verifier V chooses *a* and *b* at random from Z_q and sends $c = m^a g^b$ to a signer.

(step 2) The signer P chooses *r* at random from Z_q and sends $h_1 = c \cdot g^r$ and $h_2 = h_1^s$ to V.

(step 3) V sends *a* and *b* to P.

(step 4) P checks whether *c* is constructed properly using *a* and *b*. If the check is successful, P reveals *r* to V.

(step 5) Using *r*, *y* and *z*, V checks h_1's validity and whether

$$h_2 = z^a y^{b+r}.$$

(disavowal protocol)

(step 1) V chooses $t \in \{0, \cdots, k\}$ and $a \in Z_q$ independently and uniformly. Then V computes

$$A = m^t g^a, \ B = z^t y^a$$

and sends them to P.

(step 2) (1) P determines the value of *t* by trail and error. An efficient approach for this raises the first component *A* of the message to the *s* power and forms a quotient with the second component *B*. The *k+1* trial quotients can then be computed each by a single multiply from the quotient of the valid signature with z.

(2) P sends *blob(r,t)* which is a commitment of t using random number r.

(step 3) V sends a to P.

(step 4) P checks that a can be used to reconstruct the first message. After that, P provides r to V.

(step 5) V reveals t from *blob(r,t)* using r and checks whether it is identical to the one he chose at step 1.

4. Secret Sharing Scheme

1) Secret Sharing Scheme and Lagrange Interpolation Formula

In a (k,n) threshold scheme, a secret s is divided into "shares" $v_{1,}\cdots,v_n$ in such a way that

(a) any k shares are sufficient to efficiently reconstruct s and

(b) any k-1 shares provide no more information about the value of s.

Shamir's (k,n) threshold scheme[14] is as follows: Let $s\in_R\{1,2,\cdots,S\}$.

1. A dealer(D), whose secret is s, chooses a prime q such that $q \geq \max(S,n+1)$.
2. D chooses at random a polynomial $f(x)$ of degree at most k-1 such that
$$f(x)=f_0+f_{1x}+\cdots+f_{k-1}x^{k-1} \bmod q$$
where $f(0)=f_0=s$.
3. D gives $v_i=f(i)$ to a participant P_i for $1\leq i\leq n$.

Let $\{i_{1,}\cdots,i_k\}\subset\{1,\cdots,n\}$. $f(x)$ is reconstructed from $v_{i_1},\cdots,v_{i_k}$ by using Lagrange interpolation formula as follows.

$$f(x)=\sum_{j=1}^{k}\prod_{\Lambda not=j}\frac{x-i_l}{i_{j-i_l}}v_{i_j} \quad \bmod q. \qquad (2)$$

Then s can be computed as follows.

$$s=f(0)=\sum_{j=1}^{k}b_jv_{i_j} \quad \bmod q \qquad (3)$$

where

$$b_j \overset{\triangle}{=} \prod_{l\neq j}\frac{-i_l}{i_{j-i_l}} \quad \bmod q. \qquad (4)$$

2) Non-Interactive Verifiable Secret Sharing

In a (k,n) threshold scheme, if the dealer does not distribute the correct $\{v_i\}_{i=1,\cdots,n}$, k participants cannot obtain the secret s. Therefore, the dealer is sometimes required to prove in zero-knowledge that $\{v_i\}_{i=1,\cdots,n}$ is correct. Such a protocol is called a verifiable secret sharing scheme (VSS). [8] showed a non-interactive VSS. [13] modified it slightly. We show a further modification of [13].

(1) D publishes $GF(2^n)$ such that $2^{n-1}=q$, where q is a prime. He also publishes a primitive element g of $GF(2^n)$.

(2) The dealer D executes the Shamir's (k,n) threshold scheme (see 2.4.1).

Hereafter, calculations are executed over $GF(2^n)$.

(3) D computes $F_j \overset{\triangle}{=} g^{f_j}$ for $j=0,1,\cdots,k-1$. D publicizes $\{F_j\}_{j=0,1,\cdots,k-1}$.

(4) Note that

$$\begin{aligned}\prod_{j=0}^{k-1}F_j^{i^j} &= \prod_{j=0}^{k-1}g^{f_ji^j}\\ &= \underset{g}{\sum_{j=0}^{k-1}}f_ji^{j\,bmod}q\\ &= g^{f(i)}\\ &= g^{v_i}.\end{aligned}$$

The second line comes from the fact that $g^q=1$.

P_i verifies that $\prod_{j=0}^{k-1}F_j^{i^j}=g^{v_i}$. If this check holds, P_i is convinced that his share is certainly $f(i)$.

3) How k out of n Share Holders Compute a^s over GF(2n)

Suppose that everyone agrees on $GF(2^n)$ such that $2^{n-1}=q$, where q is a prime. Suppose also that P_i has v_i as his share for a secret s ($1 \leq i \leq$ n) (see 2.4.1). On common input λ, any k members $P_{i_1},\cdots,P_{i_k}$ can compute λ^s over $GF(2^n)$ without revealing s(and v_i) as follows.

Each P_{i_j} computes $\lambda^{b_jv_{i_j}}$ over $GF(2^n)$ and broadcasts it (b_j is defined by eq. (4). Then each P_{i_j} can compute

$$\begin{aligned}\prod_{j=1}^{k}\lambda^{b_jv_{i_j}} &= \underset{\lambda}{\sum_{j=1}^{k}}b_jv_{i_j} \bmod q\\ &= \lambda^s \text{ over } GF(2^n)\end{aligned}$$

III. Related Works

1. Undeniable Signature Scheme with Distributed Provers

Pedersen [12] showed how the signer of undeniable signatures can distribute part of his secret key to n agents such that any k of these can verify a signature. His scheme is based on Boyar et al.'s convertible undeniable signature scheme [1].

Convertible Undeniable Signature Scheme:

Convertible undeniable signatures are particular undeniable signatures which have a nice property that the signer can convert all the undeniable signatures to ordinary signatures by releasing a part of the secret key, and selectively convertible undeniable signature allows the signer to convert only selected undeniable signatures to ordinary signatures without affecting other undeniable signatures. Let p and q be large primes that q divides p-1, and g be a generator of the subgroup, G_q, of Z_p of order q.

(secret keys) $x,\ z \in Z_q^*$

(public key) (p, q, g, y, u), where $y = g^x$ and $u = g^z \bmod p$

(plaintext) m

(signature) (g^t, r, s),

where (r,s) is the ElGamal signature on $M = g^t tzm \bmod q$. That is, $g^M = y^r r^s$.

(verification)

Given m and (T,r,s), both the signer(S) and the verifier(V) can compute $w = T^{Tm}$ and $v = y^r r^s$.

(step 1) V chooses $a, b \in Z$ and sends $ch = w^a g^b$.

(step 2) S chooses $r \in Z$ and sends $h_1 = ch^r$ and $h_2 = h_1^z$.

(step 3) V sends a and b.

(step 4) S verifies that $ch = w^a g^b$. If the check is successful, S sends r to V.

(step 5) V verifies that $h_1 = (w^a g^b)^r$ and $h_2 = (v^a u^{b)^r}$.

The signer can convert all his signatures to ordinary signatures by releasing z. Alternatively, a signature (T,r,s) on the message m can be converted to a digital signature by releasing t such that $T = g^t$. Given t, a signature can be verified as follows:

1. Verify that $T = g^t$.
2. Verify that $(u^{mT})^t = y^r r^s$.

Distributed verification:

Consider the case where the signer S have signed the message m using the random exponent t. That is, the signature on m is (T,r,s) where $T = g^t$ and (r,s) is the ElGamal signature on $Ttzm$ bmod q. S distributes the ability to verify this signature to n agents $P_1, \cdots, P_n$.

(step 1) S broadcasts T to the n agents.

(step 2) S distributes t using Shamir's secret sharing scheme. Thus, P_i gets the share $t_i = f(x_i)$, where f is a polynomial over Z_q of degree k-1 such that $f(0)=t$.

(step 3) S sends H(m,r,s) to each agent, where H is a collision-free hash function. After the execution of this protocol(Before executing step 3, each participant must verify his share.), each P_i has a secret share t_i with corresponding public information $h_i = g^{t_i}$.

When a person V asks k agents (say $P_1, \cdots, P_k$) to verify a signature (T', r', s') on a message m', these k agents must check that it is the one they should verify. Let $a_1, \cdots, a_k$ satisfy the next equation.

$$t = \sum_{i=1}^{k} a_i t_i.$$

(step 4) V and each P_i verify that

$$T' = \prod_{i=1}^{k} h_i^{a_i}.$$

If this equation fails, they cannot neither verify nor deny the signature.

(step 5) Each P_i checks that the signer has sent $H(m', r', s')$. If this is true, the agents agree to verify the signature and otherwise they tell V that they are not able to verify it.

Next, $P_1, \cdots, P_k$ verify a signature by executing next protocol.

(step 6) P_i and V compute $w = u^{Tm'}$ and $v = y^{r'} r'^{s'}$.

(step 7) V chooses $a, b \in Z$ and sends $ch = w^a g^b$.

(step 8) P_i chooses $r_i \in Z$ and sends $h_{i1} = ch^{r_i}$ and $h_{i2} = h_{i1}^{t_i}$.

(step 9) V sends a and b.

(step 10) P_i verifies that $ch = w^a g^b$. If the check is successful, each P_i sends r_i to V.

(step 11) V verifies that $h_{i1} = (w^a g^b)^{r_i}$ and that

$$\prod_{i=1}^{k} h_{i2}^{a_i/r_i} = v^a T^b.$$

(step 12) V accepts the signature if and only if it accepts the proof.

2. (n,n) Undeniable Signature Scheme

An (n,n) undeniable signature scheme was shown by [11]. In this scheme, a signer is a group of n members $A_1, \cdots, A_n$. Each role of the signer is played by the cooperation of all n members.

Set Up Phase:

(1) Some member A_j publicizes a large prime p and a primitive element g of $GF(p)$.

(2) A_i $(1 \leq i \leq n)$ chooses x_i such that $\gcd(x_i, p-1) = 1$. x_i is the secret key of A_i.

Public Key Generation Phase:

The group public key is given by p,g and y such that

$$y = {}_g\prod_{i=1}^{n} x_i \mod p.$$

By the following protocol, $\{A_i\}_{i=1,\cdots,n}$ compute y cooperatively while keeping x_i secret. We use a convention such that $A_i = A_{(i \bmod n)}$ and $y_{i,t} = y_{((i \bmod n),t)}$.

(step 1) Each A_i calculates

$$y_{i,1} = g^{x_i} \quad \text{bmod } p$$

and transmits this value to A_{i+1}.

(step 2) For t=2 to n-1,

each A_i receives $y_{i-1,t-1}$ from A_{i-1} and uses his secret key x_i to compute

$$y_{i,t} = (y_{i-1,t-1})^{x_i} \mod p.$$

The result $y_{i,t}$ is transmitted to A_{i+1}.

(step 3) Each A_i receives $y_{i-1,n-1}$ from A_{i-1} and computes

$$y_{i,n}=(y_{i-1,n-1})^{x_i} \mod p.$$

Then each A_i opens $y_{i,n}$ to all group members.

(step 4) Each A_i checks that all $y_{j,n}$ $(j=1,\cdots,n)$ are equal.

If so, y is given by

$$y=y_{1,n}=\cdots=y_{n,n}={}_{g}\prod{}_{i=1}^{n}x_i \mod p.$$

y is published.

Signature Issuing Phase:

For a message m, the undeniable signature Z is defined as

$$Z={}_{m}\prod{}_{i=1}^{n}x_i \mod p.$$

$\{A_i\}_{i=1,\cdots,n}$ computes Z cooperatively as they do in the group public key generation phase.

Confirmation Protocol:

From the confirmation protocol of 2.2, we see that $\{A_i\}_{i=1,\cdots,n}$ have only to calculate ${}_{w}(\prod{}_{i=1}^{n}x_i)^{-1} \mod p$ for verifier's challenge w. Now, because

$${}_{w}(\prod_{i=1}^{n}x_i)^{-1} \mod p={}_{w}\prod_{i=1}^{n}x_i^{-1} \mod p,$$

this calculation is executed as in the group public key generation phase, using $\{x_i^{-1}\}_{i=1,\cdots,n}$ instead of $\{x_i\}_{i=1,\cdots,n}$, where x_i^{-1} is the multiplicative inverse of x_i mod p-1.

The disavowal protocol is similar.

IV. Framework

1. Outline

All signature schemes shown in section 2 are based on the discrete log problem. Each scheme has a public key of the form $y=g^s$. In this paper, for each of the schemes, we show a group (k,n) threshold signature scheme such as follows.

(1) We denote by $P_1,\cdots,P_n$ a group of n signers.

(2) No trusted dealer is necessary.

(3) All members agree on $GF(2^n)$ and g, where $2^{n-1}(\triangleq q)$ is a prime and g is a primitive element of $GF(2^n)$. Instead of $GF(2^n)$, we can use GF(p) such that p-1 is divided by a large prime q. In this case, let G_q be the subgroup of GF(p) of order q and let g be a generator of G_q.

(4) Each of the proposed schemes consists of a public key generation phase and a signature issuing phase. In the public key generation phase, all n signers must cooperate. In the signature issuing phase, any subset of k signers can issue a signature but k-1 signers can't. In undeniable signature schemes, (k,n) threshold confirmation protocol and (k,n) threshold disavowal protocol are also shown.

In the next subsection, we show a public key generation phase which is commonly used in all our group signature schemes. In the public key generation protocol, $y(=g^s)$ is produced as a public key so that nobody knows s. Each P_i obtains $v_i(=f(i)$ mod $q)$ secretly, where $f(x)$ is a random polynomial of order k-1 such that $f(0)$ = s mod q.

2. Public Key Generation Phase

Let $C(u,r)$ denote a commitment of u, where r is a random number.

(step 1) P_i chooses $x_i \in Z_q$ at random and computes $y_i=g^{x_i}$ over $GF(2^n)$. Then a random string r_i is chosen and P_i broadcasts $C_i \triangleq C(y_i, r_i)$ to all members.

(step 2) After all n signers have broadcast the commitments, each P_i opens C_i.

(step 3) The public key y is computed by

$$y=\prod_{i=1}^{n}y_i$$

over $GF(2^n)$.

Let

$$s \triangleq \sum x_i \mod q \tag{5}$$

Then,

$$y=g^s \text{ over } GF(2^n) \tag{6}$$

It is clear that nobody knows s.

Next s is distributed to $P_1,\cdots,P_n$ in a (k,n) threshold scheme sense.

(step 4) P_i chooses at random a polynomial $f_i(z)$ of degree k-1 over GF(q) such that

$$f_i(0)=x_i. \tag{7}$$

Let

$$f_i(z)=f_{i,0}+f_{i,1}z+\cdots+f_{i,k-1}z^{k-1}$$

where $f_{i,0}=x_i$.

(step 5) P_i computes $F_{i,j}=g^{f_{i,j}}$ over $GF(2^n)$ for $j=0,\cdots,k-1$ and broadcasts $(F_{i,j})_{j=1,\cdots,k-1}$.

(step 6) After every signer have sent these k-1 values, P_i sends

$$s_{i,j}=f_i(j) \tag{8}$$

secretly and a signature on $s_{i,j}$ to P_j for $j=1,\cdots,n$.

(step 7) P_i verifies that the share received from P_j, $(S_{j,i})$ is consistent with the previously published values by verifying that

$$g^{s_{j,i}}=\prod_{l=0}^{k-1}F_{j,l}^{i^l} \text{ over } GF(2^n).$$

If this fails, P_i broadcasts that an error has been found, publishes $s_{i,j}$ and the signature (since P_i falsely claims to have not found the error) and then stops.

(step 8) P_i computes his share of s, v_i, as the sum of all shares received in step 6:

$$v_i = \sum_{j=1}^{n} s_{j,i} \mod q \tag{9}$$

lemma 4.1 *In the above protocol*

(1) $y = g^s$.

(2) $v_i = f(i)$ for some polynomial $f(z)$ of order k-1 such that $f(0)=s$.

proof 1 (1) is clear.

(2) Let f be $f(z) = f_1(z) + \cdots + f_n(z)$. Then, $f(0)=s$ from eq. (5) and eq. (7). From eq. (8) and eq. (9), $v_i = f(i)$.

V. Proposed (k,n) Threshold Digital Signature

1. Modified ElGamal Signature Scheme

We first modify ElGamal signature scheme [7]. This modification is very suitble for our purpose, cooperation based signature schemes.

Let g be a primitive element of $GF(p)$ and p-1 has a large prime factor. Let h be a one-way hash function.

(secret key)	s (0<s<p-1)
(public key)	$y(=g^s)$
(plaintext)	m
(signature generation)	Choose $r \in Z_{p-1}$ at random}. Let $e = g^r$ mod p and $z=h(m)*r-se$ mod p-1. ($h(m) \neq$ 0, p-1/2).
(digital signature)	(e,z)
(verification)	$e^{h(m)} = g^z y^e \mod p$

2. (k,n) Threshold Digital Signature Scheme

In this section, we show the signature issuing phase of our group digital signature scheme. The underlying digital signature scheme is the modified ElGamal scheme proposed in 5.1. (It should be noted that a group fail stop signature scheme can be obtained in a similar manner to the group digital signature scheme by using [10] as the underlying fail stop signature scheme.)

It is clear that k-1 signers cannot issue a signature because s is distributed to n signers by a (k,n) threshold scheme in the public key generation phase. We show that any k signers can issue a signature. Suppose that $P_{i_1}, \cdots, P_{i_k}$ issue a signature for a plaintext m.

(step 1) Each P_{i_j} computes $x_j(= g^{r_j}$ over GF$(2^n))$, where r_j is a random number, and broadcasts x_j.

(step 2) Let

$$e \triangleq x_1 \cdots x_k \ (= {}_g\textstyle\sum r_j). \tag{10}$$

(step 3) Each P_{i_j} opens z_j such that

$$z_j = h(m) r_j - e b_j v_{i_j} \ (\mathrm{mod}\, q), \tag{11}$$

where $b_j \triangleq \prod_{l \neq j}^{k} \frac{-i_l}{i_j - i_l}$ mod q. Remember that $v_{i_j} = f(i_j)$ (see lemma 4.1).

(step 4) Let $z = z_1 + \cdots + z_k$ mod q.

The group signature is (e,z).

lemma 5.1

$$e^{h(m)} = g^z y^e \tag{12}$$

proof 2 $f(x)$ is reconstructed from $v_{i_1}, \cdots, v_{i_k}$ by using Lagrange interpolation formula as follows [5].

$$f(x) = \sum_{j=1}^{k} \prod_{\wedge not = j} \frac{x - i_l}{i_{j-i_l}} v_{i_j} \mod q. \tag{13}$$

Then s can be computed as follows.

$$s = f(0) = \sum_{j=1}^{k} b_j v_{i_j} \mod q \tag{14}$$

Now,

$$\begin{aligned} z &= z_1 + \cdots + z_k \\ &= \textstyle\sum (h(m) r_j - e b_j v_{i_j}) \\ &= h(m)(\textstyle\sum r_j) - e(\textstyle\sum b_j v_{i_j}) \\ &= h(m)(\textstyle\sum r_j) - es \end{aligned}$$

Therefore, $g^z = g^{h(m)\sum r_j} g^{-es} = e^{h(m)} / y^e$ from eq.(10) and lemma 4.1.

The verifier verifies eq. (12).

VI. Group Fail-Stop Signature Scheme

In this section, we show the public key generation and the signature issuing phase of a group fail-stop signature scheme. We use the scheme of [10] in 2.1 as the original scheme.

Public key generation phase:

(step 1) Each member $P_i (i = 1, 2, \cdots, n)$ selects his private secret s_{1i}, s_{2i}, s_{3i} and s_{4i} randomly.

(step 2) P_i broadcasts $y_{1i} = g_1^{s_{1i}} g_2^{s_{2i}}$ and $y_{2i} = g_1^{s_{3i}} g_2^{s_{4i}}$. Then, all members compute the product of these values.

The generated public keys (y_1, y_2) are as follows.

$$y_1 = \prod_{i=1}^{n} y_{1i} = {}_{g_1}\textstyle\sum_{i=1}^{n} s_{1i} \sum_{i=1}^{n} s_{2i}$$

$$y_2 = \prod_{i=1}^{n} y_{2i} = {}_{g_1}\sum_{i=1}^{n} s_{3i}\, {}_{g_2}\sum_{i=1}^{n} s_{4i}.$$

Each member shares $s_l (l=1,2,3,4)$ by using step4 ~ 8 in 4.2. Here, $s_l \triangleq \sum_{i=1}^{n} s_{li}$.

Signature issuing phase:

Suppose that the share of P_i for s_l be $t_{li} (i=1,2,3,4)$ and $P_1, \cdots, P_k$ issue a signature on a message m.

(step 1) P_j computes

$$z_{1j} = (t_{1j} + mt_{3j}) b_j \mod p-1$$

$$z_{2j} = (t_{2j} + mt_{4j}) b_j \mod p-1$$

where b_j is Lagrange constant. Then P_j opens them.

(step 2) The group signature (z_1, z_2) is as follows:

$$\begin{aligned} z_1 &= \sum_{j=1}^{k} z_{1j} \\ &= \sum_{j=1}^{k} b_j t_{1j} + \sum_{j=1}^{k} m b_j t_{3j} = s_1 + ms_3 \\ z_2 &= \sum_{j=1}^{k} z_{2j} \\ &= \sum_{j=1}^{k} b_j t_{2j} + \sum_{j=1}^{k} m b_j t_{4j} = s_2 + ms_4 \end{aligned}$$

VII. Group Undeniable Signature Schemes

1. ZK Group Undeniable Signature Scheme

In this subsection, we show how to convert the zero knowledge undeniable signature scheme [2] in 2.3 to a (k,n) threshold version. For simplicity, we show the signature issuing phase only. The group confirmation protocol and the group disavowal protocol are derived easily. For a plaintext m, any k members $P_{i_1}, \cdots, P_{i_k}$ can compute the signature $z = m^s$ over $GF(2^n)$ without revealing s as follows.

Each P_{i_j} computes $m^{b_j v_{i_j}}$ over $GF(2^n)$ and broadcasts it. Then everyone can compute

$$\begin{aligned} \prod_{j=1}^{k} m^{b_j v_{i_j}} &= {}_m\Sigma_{j=1}^{k} b_{j v_{i_j} \bmod q} \\ &= m^s (\text{ from } eq. \text{ (14).)} \end{aligned}$$

2. 2 Move Group Undeniable Signature Scheme

It is not easy to convert a 2 move undeniable signature scheme of 2.2 to a (k,n) threshold version. At step 2 of the confirmation protocol in 2.2, the signer P must compute $R = w^{s^{-1}}$ (not w^s). When each P_i has a share of s, it is easy for k out of n share holders to compute λ^s cooperatively. However, we don't have such an efficient multiparty protocol for computing $\lambda^{s^{-1}}$. To avoid this problem, we modify the confirmation protocol so that P computes $R = w^s$ at step 2. The modified confirmation protocol is as follows.

(step 1) V randomly selects two integers a and b, from Z_q, and computes

$$w = m^a g^b. \tag{15}$$

w is sent to P.

(step 2) P computes

$$R = w^s.$$

R is sent back to V.

(step 3) V checks whether

$$R = z^a y^b.$$

If this equality holds, the signature has been verified. The disavowal protocol is modified similarly. Now, a (k,n) threshold version of this protocol is obtained by using the same method in 7.1.

1) Security

Next, we show the security of the proposed scheme.

The signers' fraud are the following two cases.

(1) For the invalid signature, the signers convince the verifier that it is valid in the confirmation protocol.

(2) For the valid signature, the signers convince the verifier that it is invalid in the disavowal protocol.

It is easy to see Proposition 7.1.

proposition 7.1 There exist q pairs, satisfying eq. (15), $(a_1, b_1), \cdots, (a_q, b_q)$ where $a_i \neq a_j, b_i \neq b_j$ for $i \neq j$.

Hereafter, calculations are executed over $GF(2^n)$. Let

$$L(w) \triangleq \{(a, b) \mid w = m^a g^b\}$$

proposition 7.2 If $Z \neq m^x$, $(a_i, b_i) \in L(w)$, $(a_j, b_j) \in L(w)$ and $(a_i, b_{i)} \neq (a_j, b_j)$, then

$$Z^{a_i} y^{b_i} \neq Z^{a_j} y^{b_j}.$$

Proof 3 Since $2^{n-1} = q$ and q is a prime, $Z = m^{x'}$ for some $x' \neq x$ if $Z \neq m^x$ over $GF(2^n)$. From the assumption, we have

$$\begin{aligned} m^{a_i} g^{b_i} &= m^{a_j} g^{b_j}, \\ m^{a_i - a_j} &= g^{b_j - b_i}. \end{aligned} \tag{16}$$

Suppose that

$$Z^{a_i} y^{b_i} = Z^{a_j} y^{b_j}.$$

Then, we have

$$\begin{aligned} (m^{x'})^{a_i} (g^x)^{b_i} &= (m^{x'})^{a_j} (g^x)^{b_j}, \\ (m^{a_i - a_j})^{x'} &= (g^{b_j - b_i})^{x}. \end{aligned} \tag{17}$$

From eq.(16) and (17),

$$(g^{b_j - b_i})^{x' - x} = 1.$$

Because of $b_j \neq b_i, x = x'$. This is a contradiction.

Theorem 7.1 Even if signers have unlimited computing power, the probability providing a valid response for an invalid signature is at most $\frac{1}{q}$.

proof 4 We can regard $S(\triangleq(A_{i_1},\cdots,A_{i_k}))$ and V as two probabilistic Turing Machines. Let t_S, t_V random tapes of S and V, respectively. t_V is used for (a,b) and t_S is used to compute R. Then we prove that $\forall S$, $\forall p$, $\forall g$, $\forall y$, $\forall m$, $\forall Z$ (*invalid signature for m*)

$$\Pr_{t_S, t_V}[\text{ S provide a valid response}] \le \frac{1}{q}. \qquad \textbf{(18)}$$

At first, we define t_V^i as follows:

$$t_V^i \triangleq \{t \in t_V \mid V(t) \in L(w_i)\},$$

where $V(t)$ is V's selection of (a,b) when it uses t as its random tape. Assume that

$$\Pr_{t_S, t_V^i}[\text{ S provide a valid response}] > \frac{1}{q}.$$

Then $\exists t_{S0} \in t_S$ such that

$$\Pr_{t_V^i}[\text{ S provide a valid response}] > \frac{1}{q}.$$

Because a honest V selects (a,b) randomly,

$$\exists t_{S0}, \Pr_{L(w_i)}[\text{ S provide a valid response}] > \frac{1}{q}.$$

From Proposition 7.1, $|L(w_i)| = q$. So, this means that there exist two pairs $(a_j, b_j), (a_k, b_k) \in L(w_i)$ such that if S send a response R_0 corresponding to t_{S0}, V accepts R_0 in both case that V selects (a_j, b_j) and (a_k, b_k). This contradicts Proposition 7.2. That is,

$$\Pr_{t_S, t_V^i}[\text{ S provide a valid response}] \frac{1}{q}.$$

We can show this for all $i(=1, \cdots, q)$, so (18) is proved.

Theorem 7.2 Even if signers have unlimited computing power, the probability that the signers can disavow the valid signature is at most $\frac{1}{q}$.

proof 5 In the disavowal protocol, V checks whether

$$(Ry^{-b})^c = (R'y^{-d})^a.$$

That is, in order to disavow the valid signature, the signers have to send R' such that

$$R' = \{(Ry^{-b})^{1/a}\}^c y^d.$$

Suppose that in the worst case, the signers have known the values of a and b. Then $(Ry^{-b})^{1/a}$ is known to the signers. So, if the signers can send such a response R' with a probability better than $\frac{1}{q}$, it contradicts Theorem 7.1.

Remark 1 In 7.2.1, we showed only the security of the signer's fraud of the modified scheme. However, since Chaum's 2 move undeniable signature and the modified undeniable signature scheme are not zero-knowledge[9], malicious verifiers can generate undetectable forgeries of signer's signature. Thus, ZK scheme [2] must be used for undeniable signature.

VIII. Conclusion

In this paper, we have shown (k,n) threshold signature versions based on the discrete logarithm problem: a threshold digital signature, a threshold fail-stop signature and two threshold undeniable signatures, respectively. Furthermore, we have presented a modified ElGamal signature scheme to make it suitable for group use.

Acknowledgement

The author would like to thank Professor Kaoru Kurosawa of Tokyo Institute of Technology for his several helpful comments. The author also would like to thank anonymous referees for their valuable comments on the manuscript of this paper.

References

[1] J. Boyar, D. Chaum, I. B. Damgard and T. P. Pedersen, "Convertible Undeniable Signatures," *Advances in Cryptology, Proceedings of Crypto'90*, pp. 189-205, 1991.

[2] D. Chaum, "Zero-Knowledge Undeniable Signatures" *Advances in Cryptology, Proceedings of Eurocrypt'90*, pp. 458-464, 1990.

[3] D. Chaum and H. V. Antwerpen, "Undeniable Signatures" *Advances in Cryptology, Proceedings of Crypto'89*, pp. 212-216, 1990.

[4] Y. Desmedt, "Society and Group Oriented Cryptography: A New Concept" *Advances in Cryptology, Proceedings of Crypto'87*, pp. 120-127, 1988.

[5] Y. Desmedt and Y. Frankel, "Threshold Cryptosystem" *Advances in Cryptology, Proceedings of Crypto'89*, pp. 307-315, 1990.

[6] Y. Desmedt and Y. Frankel, "Shared Generation of Authenticators and Signatures" *Advances in Cryptology, Proceedings of Crypto'91*, pp. 457-469, 1991.

[7] T. ElGamal, "A Public Key Cryptosystem and a Signature Scheme based on Discrete Logarithm" *IEEE Trans.*, Vol. IT-31, No. 4, pp. 469-472, 1985.

[8] P. Feldman, "A Practical Scheme for Non-Interactive Verifiable Secret Sharing" *Proceedings of the 28th IEEE Symposium on the Foundations of Computer Science*, pp. 427-437, 1987.

[9] O. Goldreich and H. Krawczyk, "On the composition of zero-knowledge proof systems" *Proceedings of the 17th annual International Colloquium on Automata, Languages and Programming*, pp. 268-282, 1990.

[10] E. V. Heyst and T. P. Pedersen, "How to Make

Efficient Fail-Stop Signatures" *Advances in Cryptology, Proceedings of Eurocrypt'92*, pp. 366-377, 1993.

[11] L. Harn and S. Yang, "Undeniable Signature Schemes without the Assistance of a Mutually Trusted Party" *Advances in Cryptology, Proceedings of Auscrypt'92*, pp. 133-142, 1992.

[12] T. P. Pedersen, "Distributed Provers with Applications to Undeniable Signatures" *Advances in Cryptology, Proceedings of Eurocrypt'91*, pp. 221-238, 1991.

[13] T. P. Pedersen, "A Threshold Cryptosystem without a Trusted Party" *Advances in Cryptology, Proceedings of Eurocrypt'91*, pp. 522-526, 1991.

[14] A. Shamir, "How to Share a Secret" *Communication of the ACM*, Vol. 24, pp. 612-613, 1979.

Choonsik Park received the B.S. degree form Kwangwoon University and the M.S. degree from Hanyang University, Seoul, Korea in 1981 and 1983, respectively, and the Dr. Eng. degree in electronic engineering from Tokyo Institute of Technology, Tokyo, Japan in 1995, Since joining Coding Technology and Research Section in Electronics and Telecommunications Research Institute (ETRI) in 1982, he has been engaged in research and development on information security. His research interests are information security and cryptographic protocols.