Threshold Ring Signature Scheme based on the Curve

Kuwakado, Hidenori

Tanaka, Hatsukazu

Vol. 44 No. 8 IPSJ Journal Aug. 2003

Regular Paper

Threshold Ring Signature Scheme Based on the Curve

HIDENORI KUWAKADO† and HATSUKAZU TANAKA†

Rivest, Shamir, and Tauman have proposed the ring signature scheme, which makes it possible to specify a group without revealing which member signed a message. Bresson, Stern, and Szydlo have shown a (k, n) threshold ring signature scheme. It is possible to convince a verifier that at least k members in the n-member group signed a message without revealing which k members signed the message. In this paper, we propose a new (k, n) threshold ring signature scheme. While the previous schemes form a ring of individual signatures, our scheme forms a curve of individual signatures. Our scheme is more efficient than the scheme shown by Bresson, et al. Moreover, we show that ElGamal's signature scheme, which is not based on the trapdoor one-way permutation, is available in the threshold ring signature.

1. Introduction

Digital signature schemes are primitive tools to achieve authenticated communication. Since the concept of digital signatures was proposed by Diffie and Hellman[4], a lot of effort has been devoted to achieving secure digital signature schemes. As a result, digital signatures with remarkable properties have been developed. For example, group signatures have the following properties[3]: (i) a verifier can verify that one of group members signed a message, (ii) the verifier cannot discover which group member signed it. It seems difficult to achieve such properties with handwriting signatures. Thus, digital signatures are superior to handwriting signatures from the viewpoint of functional respects.

Rivest, et al.[9] have proposed a concept of *ring signatures*, and they have shown its implementation (called the *RST scheme*). The ring signature is one of group signatures. The ring signature enables a signer to make a group signature without special preparation and other members' cooperation. The ring signature provides an elegant way to leak authoritative secrets anonymously. The ring signature assumes that the number of signers is one.

Bresson, et al.[1] have proved that a secure ring signature is constructable without an ideal cipher. Moreover, they have shown a (k, n) threshold ring signature scheme (called the *BSS* scheme); k and n are the number of signers and that of group members respectively, the verifier can verify the message and the number of signers, and the verifier cannot know which k group members signed the message. In the case of leaking secrets, information on the number of signers is important for the verifier because more group members sign the message, more credible the message is.

In this paper, we propose a new (k, n) threshold ring signature scheme. While the RST scheme and the BSS scheme geometrically make a ring of individual signatures, the proposed scheme makes a curve of individual signatures. The proposed scheme is more efficient than the BSS scheme. Moreover, we show that ElGamal's signature scheme is available in the threshold ring signature.

1.1 Related Work

The properties of the ring signature are summarized below[9]. The ring signature assumes that the number of signers is one.

(1) The ring signature makes it possible to specify a set of possible signers without revealing which member signed a message. Namely, the ring signature is signer-ambiguous.

(2) The ring signature is setup-free; the signer does not need the knowledge, consent, or assistance of the other members. All the signer needs is knowledge of their regular public keys.

The notion of ring signatures is not completely new, but previous references did not explicitly formalize the notion, and proposed inefficient constructions[9]. For example, Camenisch's scheme[2] may be viewed as a ring signature scheme.

The RST scheme has the above two properties. The RST scheme is two to three orders of magnitude faster than Camenisch's scheme. The RST scheme requires all the members to use signature schemes based on trapdoor

† Kobe University

one-way permutations, e.g., the RSA signature scheme[8].

The BBS scheme is the (k, n) threshold version of the RST scheme. The BBS scheme utilizes a super ring and sub rings. The super ring proves that at least one split that consists of sub rings has been solved. For the other splits, one has to simulate a correct ring signature for every unsolved sub ring. In order to generate proper sub rings, the signers use the complete partitioning system. Thus, the BSS scheme is the natural extension of the RST scheme. The objective of the BSS scheme is same as that of our scheme. Hence, we mainly compare our scheme with the BSS scheme with respect to the efficiency.

Naor[6] has combined deniable authentications and ring signatures, and achieved *deniable ring authentications*, which have the deniablity and the signer anonymity. The efficiency of deniable ring authentications is comparable to that of ring signatures (within multiplicative constant). Naor discussed the extension of the number of signers; Naor's scheme can authenticate the statement that at least k signers signed the message.

Since ring signature schemes use individual regular signature schemes, ring signature schemes extend domains of the individual verification functions to a large common domain. Ohkubo, et al.[7] have constructed ring signature schemes without the domain extension by using hash functions (called the *OASI schemes*). The OASI schemes assume that the number of signers is one.

1.2 Contribution of This Work

In this paper, we propose a new (k, n) threshold ring signature scheme. The comparison with previous schemes are summarized below.

Comparison with the RST scheme The RST scheme enables the verifier to verify that *at least one* ring member signed the message. Even if two ring members signed the message, the verifier cannot verify the number of signers because of the structure of the ring. In the case of leaking secrets, information on the number of signers is important for the verifier. Our scheme enables the verifier to verify that *at least* k ring members signed the message. In order to achieve a (k, n) threshold ring signature scheme, we use a curve instead of the ring. The construction of our scheme is applicable to not only the RSA scheme but also ElGamal's signature scheme[5]. The disadvantage of our scheme is the speed of the signing and the verification. Our scheme is at least two times as slow as the RST scheme.

Comparison with the BSS scheme The objective of the BSS scheme is similar to that of our scheme. The BSS scheme is a ring signature scheme with ad-hoc access structure including the (k, n) threshold access structure. However, their constructions are completely different; while the BSS scheme makes plural rings of individual signatures, our scheme makes the curve of individual signatures. Our scheme is more efficient than the BSS scheme as follows. The size of a signature of our scheme is $O(n)$, and that of the BSS scheme is $O(2^k n \log n)$. With respect to the number of computations of individual signatures, our scheme is $O(n)$, and the BSS scheme is $O(2^k n \log n)$. Specifically, when a modulus of the RSA function is a 1024-bit number, its exponent is 3, and $k \geq 10$, the time for generating a signature of our scheme is shorter than that of the BSS scheme. The time for verifying a signature of our scheme is always shorter than that of the BSS scheme.

Comparison with Naor's scheme Since Naor's scheme is interactive, some mechanism of anonymous routing (e.g., MIX-net) is required for achieving the anonymity. Since our scheme is non-interactive, our scheme does not require such a mechanism. Our construction is completely different from Naor's construction.

Comparison with Camenisch's scheme Our scheme is setup-free. Namely, the signers do not need the knowledge, consent, or assistance of non-signers. All the signers need is knowledge of regular signature schemes of non-signers. Camenisch's scheme is not completely setup-free. Although Camenisch's scheme is based on the discrete logarithm problem, our scheme can use not only the discrete logarithm problem (DLP) but also the factoring problem as the basis problem.

Comparison with the OASI scheme While the OASI schemes are $(1, n)$ threshold ring signature schemes, our scheme is a (k, n) threshold ring signature scheme. One of the OASI schemes uses domains of individual verification functions instead of the large common domain. We show the schemes based on the domains that are smaller than any domains of the individual verification functions. In Ref. 7), the

DLP-based ring signature and the RSA-based ring signature were discussed separately. We show that ElGamal-based members and RSA-based members can exist together.

The organization of this paper is as follows. In Section 2, we describe definitions and the RST scheme. The method for extending domains of trapdoor permutations to a common domain is also used in our scheme. In Section 3, we propose a (k, n) threshold signature scheme. We prove that the anonymity of signers is unconditionally secure, and the forgery is as hard as the forgery of the individual signature schemes under the random oracle model. In Section 4, we first improve the efficiency of our scheme by decreasing the size of the common domain. Next, we propose a (k, n) threshold ring signature scheme based on ElGamal's signature scheme. In Section 5, we conclude this paper.

2. Preliminaries

2.1 Definitions

The terminology of this paper is basically same as that of Ref. 9). We call a set of possible signers A_i $(i \geq 1)$ a *ring*, denoted by $\mathcal{R}$. We call a ring member who generates a signature a *signer*, and each of the other ring members a *non-signer*. Let $\mathcal{S}$ and $\bar{\mathcal{S}}$ denote a set of signers and that of non-signers, respectively. Here, we have $\mathcal{R} = \mathcal{S} \cup \bar{\mathcal{S}}$ and $\mathcal{S} \cap \bar{\mathcal{S}} = \emptyset$. We assume that a ring member A_i is associated with a public key P_i that specifies the procedure for verifying A_i's signature.

A (k, n) threshold ring signature scheme is defined by the following procedures.

(1) A signing procedure produces a (k, n) threshold ring signature σ for a message m, given the public keys P_i $(i \in \mathcal{R})$ of n ring members and the secret keys S_i $(i \in \mathcal{S})$ of k signers.
$$\sigma = \mathsf{sign}(m, \{P_i | i \in \mathcal{R}\}, \{S_i | i \in \mathcal{S}\})$$

(2) Given m, k, and σ, a verification procedure $\mathsf{verify}(m, k, \sigma)$ outputs true if m and k are valid. Otherwise it outputs false.

The (k, n) threshold ring signature scheme authenticates the statement that at least k ring members in $\mathcal{R}$ signed the message. The $(1, n)$ threshold ring signature scheme is equivalent to the original ring signature scheme.

2.2 RST Scheme

This section mentions the RST scheme[9]. Each ring member A_i has an RSA public key P_i that specifies the trapdoor one-way permutation
$$f_i(x) = x^{e_i} \bmod N_i.$$
We assume that only A_i can compute the inverse permutation f_i^{-1} efficiently[8].

For each f_i, the extended trapdoor permutation g_i over the *common domain* $\{0,1\}^b$ is defined in the following way. For $x \in \{0,1\}^b$,
$$g_i(x) = \begin{cases} q_i N_i + f_i(r_i) & \text{if } (q_i+1)N_i \leq 2^b, \\ x & \text{otherwise,} \end{cases}$$
where $x = q_i N_i + r_i$ and $0 \leq r_i < N_i$. Notice that 2^b is much larger than any N_i's; for example, $2^b \approx 2^{160} \cdot \max_{i \in \mathcal{R}}(N_i)$.

Given a message m to be signed, the signer computes a ring signature as follows. The signer first computes a key $z = h(m)$ where h is a public collision-resistant hash function from $\{0,1\}^*$ to $\{0,1\}^\ell$. The signer uniformly chooses v from $\{0,1\}^b$ at random. The signer picks $x_i \in \{0,1\}^b$ $(i \in \bar{\mathcal{S}})$ at random, and computes $y_i = g_i(x_i)$. The signer solves the following equation for y_i $(i \in \mathcal{S})$. Notice that the number of signers is one, i.e., $|\mathcal{S}| = 1$.
$$E_z(y_n \oplus E_z(y_{n-1} \ldots \oplus E_z(y_1 \oplus v) \ldots))) = v, \quad (1)$$
where E is a symmetric encryption with a block length b and a key length ℓ. Using the trapdoor information, the signer inverts g_i on y_i for $i \in \mathcal{S}$. i.e.,
$$x_i = g_i^{-1}(y_i).$$
The ring signature of m is defined as
$$\sigma = \{v, (P_i, x_i)\ (i \in \mathcal{R})\}.$$

A verifier can verify the ring signature σ as follows. For $i \in \mathcal{R}$, the verifier computes $y_i = g_i(x_i)$. The verifier obtains z by hashing m. The verifier checks that the y_i's satisfy Eq. (1). If it is satisfied, then the verifier accepts m as valid. Otherwise the verifier rejects m.

We observe that the RST scheme authenticates the statement that at least one ring member signed the message. Even if two signers exist, the verifier cannot check the number of signers.

3. Threshold Ring Signature Scheme

We propose a (k, n) threshold ring signature scheme based on the curve over a finite field.

3.1 Tools

Similar to the RST scheme and the BSS scheme, our simplest construction assumes that ring members use trapdoor one-way permutations to generate and verify signatures. As stated in Section 2.2, each ring member A_i has an RSA public key P_i, and f_i is transformed into g_i on the common domain $\{0,1\}^b$. We can

consider $\{0,1\}^b$ as the finite field $\mathrm{GF}(2^b)$.

We assume the existence of the public symmetric encryption E with the block length b and the key length ℓ. For a fixed key z, E_z is a permutation on $\{0,1\}^b$. We assume the existence of public collision-resistant hash functions h, h' where h is a mapping $\{0,1\}^*$ to $\{0,1\}^b$ and h' is a mapping $\{0,1\}^*$ to $\{0,1\}^\ell$. In the security proof of the proposed scheme, we will consider that E_z, E_z^{-1}, h, and h' are computed by random oracles.

3.2 Protocol

3.2.1 Signing

Given a message m to be signed, k signers generate a (k,n) threshold ring signature as follows. We can assume that $1 \le k \le n-1$; if $k=0$, then there is no signer, and if $k=n$, then all ring members are signers.

step 1 The signers select $n-k$ non-signers arbitrarily. For simplicity, suppose that $\mathcal{R} = \{1,2,\ldots,n\}$.

step 2 The signers define (x_0, y_0) and z as

$$x_0 = 0,$$
$$y_0 = h(m, k, P_1, P_2, \ldots, P_n),$$
$$z = h'(m, k, P_1, P_2, \ldots, P_n).$$

step 3 The signers uniformly choose α_i $(i \in \bar{\mathcal{S}})$ from $\{0,1\}^b$ at random, and compute $x_i = E_z(g_i(\alpha_i))$. If $x_i = x_{i'}$ for $i, i' \in \bar{\mathcal{S}}$, then the signers choose α_i again. Then, the signer uniformly choose β_i $(i \in \bar{\mathcal{S}})$ from $\{0,1\}^b$ at random, and computes $y_i = E_z(g_i(\beta_i))$ for $i \in \bar{\mathcal{S}}$.

step 4 The signers determine the lowest-degree curve C that goes through the $(n-k+1)$ points (x_i, y_i) $(i \in \bar{\mathcal{S}} \cup \{0\})$.

$$C : y = c(x) = \sum_j c_j x^j, \quad c_j \in \mathrm{GF}(2^b)$$

If the degree of C is not equal to $n-k$, then the signers go back to step 3.

step 5 The signers uniformly choose $x_i \in \mathrm{GF}(2^b)$ $(i \in \mathcal{S})$ at random. If $x_i = x_{i'}$ for $i, i' \in \mathcal{S}$, then the signers choose x_i again. They compute $y_i = c(x_i)$. Using their secret keys, they compute $\alpha_i = g_i^{-1}(E_z^{-1}(x_i))$ and $\beta_i = g_i^{-1}(E_z^{-1}(y_i))$.

step 6 Finally, the (k,n) threshold ring signature of m is defined as

$$\sigma = \{C, k, (P_i, \alpha_i, \beta_i)\ (i \in \mathcal{R})\}. \quad (2)$$

In step 4, the signers obtain the curve C that goes through the $(n-k+1)$ points. The degree of C is probably $n-k$ because the probability that the degree is less than $n-k$ is 2^{-b}. Thus, the signers can probably obtain C with degree $n-k$ without going back to step 3. Since the degree of C is related with the number of signers, the signers must use C with degree $n-k$.

3.2.2 Verification

A verifier can verify the (k,n) threshold ring signature σ on the message m as follows.

step 1 The verifier checks that $1 \le k \le n-1$. If it does not hold, then the verifier rejects m and k. The verifier checks that C is an $(n-k)$-degree curve over $\mathrm{GF}(2^b)$ and each of (α_i, β_i) $(i \in \mathcal{R})$ is in $\mathrm{GF}(2^b)^2$. If one of them is not satisfied, then the verifier rejects m and k.

step 2 The verifier computes

$$z = h'(m, k, P_1, P_2, \ldots, P_n).$$

For $i \in \mathcal{R} \cup \{0\}$, the verifier computes

$$x_i = \begin{cases} 0 & \text{if } i = 0, \\ E_z(g_i(\alpha_i)) & \text{if } i \in \mathcal{R}, \end{cases}$$

$$y_i = \begin{cases} h(m, k, P_1, P_2, \ldots, P_n) & \text{if } i = 0, \\ E_z(g_i(\beta_i)) & \text{if } i \in \mathcal{R}. \end{cases}$$

step 3 For $i \in \mathcal{R} \cup \{0\}$, the verifier checks that $y_i = c(x_i)$ over $\mathrm{GF}(2^b)$. In other words, the verifier checks that all the $(n+1)$ points (x_i, y_i) $(i \in \mathcal{R} \cup \{0\})$ lie on C. If it holds, then the verifier accepts m and k. Otherwise the verifier rejects them.

3.3 Security

3.3.1 Anonymity

The identity of the signers, which the points placed later indicate, is unconditionally protected with the proposed scheme. Intuitively, the verifier cannot decide which points are placed on the pre-determined curve.

Let us consider two different signer sets $\mathcal{S}_1$, $\mathcal{S}_2$ in the same ring $\mathcal{R}$. Fix a message m to be signed. We denote by ς_i a set of possible (k,n) threshold ring signatures generated by $\mathcal{S}_i$. Any signature in ς_1 can be also generated by $\mathcal{S}_2$. Similarly, any signature in ς_2 can be also generated by $\mathcal{S}_1$. Thus, we have $\varsigma_1 = \varsigma_2$. Since the signature by $\mathcal{S}_i$ is chosen uniformly from ς_i, the distribution of signatures on ς_1 is same as that on ς_2. Therefore, the verifier cannot distinguish ς_1 from ς_2. The above discussion does not depend on any computational assumptions.

3.3.2 Forgery

First, we consider a special attack. Given a valid (k,n) threshold ring signature

$$\sigma = \{C, k, (P_i, \alpha_i, \beta_i)\ (i \in \mathcal{R})\}$$

an adversary attempts to change C and k without changing (P_i, α_i, β_i) $(i \in \mathcal{R})$. This attack intends to forge the number of signers.

Suppose that the adversary changes k to $\tilde{k}$ where $1 \leq \tilde{k} \leq n-1$. From step 2 of the verification procedure, the adversary obtains

$$\tilde{z} = h'(m, \tilde{k}, P_1, P_2, \ldots, P_n)$$

$$\tilde{x}_i = \begin{cases} 0 & \text{if } i = 0, \\ E_{\tilde{z}}(g_i(\alpha_i)) & \text{if } i \in \mathcal{R}, \end{cases}$$

$$\tilde{y}_i = \begin{cases} h(m, \tilde{k}, P_1, P_2, \ldots, P_n) & \text{if } i = 0, \\ E_{\tilde{z}}(g_i(\beta_i)) & \text{if } i \in \mathcal{R}. \end{cases}$$

Due to $E_{\tilde{z}}$, $(\tilde{x}_i, \tilde{y}_i)$ is a random point. Given the $n+1$ random points, the probability that they lie on an $(n-\tilde{k})$-degree curve is $2^{-b\tilde{k}}$. Therefore, this attack fails with the overwhelming probability.

Next, we consider a general attack. The forgery is possible if one of the individual signature schemes is insecure. Hence, we prove that the proposed scheme is as secure as the individual signature schemes. In order to do it, we will prove that Algorithm $\mathcal{A}$ that can generate a new (k, n) threshold ring signature with adaptive chosen-message attack can be transformed into Algorithm $\mathcal{B}$ that inverts one of the trapdoor one-way permutations f_i on input τ.

Algorithm $\mathcal{A}$ accepts P_i $(i \in \mathcal{R})$ and can access oracles E, E^{-1}, h, h', and a ring signing oracle. Algorithm $\mathcal{A}$ generates a valid (k, n) ring signature on a new message with non-negligible probability. Algorithm $\mathcal{B}$ uses Algorithm $\mathcal{A}$ as a black box, but fully controls oracles E, E^{-1}, h, h', and the ring signing oracle. Then, Algorithm $\mathcal{B}$ can simulate the ring signing oracle; E (and E^{-1} if necessary) is adjusted in such a way that the points (x_i, y_i) $(i \in \mathcal{R})$ lie on the specified-degree curve.

The goal of Algorithm $\mathcal{B}$ is to compute $\theta = g_i^{-1}(\tau)$ for some i and random input τ. When Algorithm $\mathcal{A}$ asks the oracle E^{-1} to answer $E_z^{-1}(x_i)$ or $E_z^{-1}(y_i)$, Algorithm $\mathcal{B}$ makes the oracle E^{-1} answer τ. If it is used in final forgery, then Algorithm $\mathcal{B}$ can find θ satisfying $\theta = g_i^{-1}(\tau)$ in the forged signature. We observe that if Algorithm $\mathcal{A}$ produces a valid (k, n) threshold ring signature, then Algorithm $\mathcal{A}$ must ask the oracle E^{-1} to answer $E_z^{-1}(x_i)$ or $E_z^{-1}(y_i)$ that is used in final forgery; if Algorithm $\mathcal{A}$ does not ask it, then all the points (x_i, y_i) in the final forgery are considered as random points because of E. When (x_i, y_i) are randomly chosen, the probability of success of the forgery is negligible.

Algorithm $\mathcal{B}$ does not know which queries on $E_z^{-1}(x_i)$ and $E_z^{-1}(y_i)$ are actually used in the final forgery. Here, Algorithm $\mathcal{B}$ makes a random guess. The probability of guessing correctly is at least $1/w$ where w is the number of queries on $E_z^{-1}(x_i)$ and $E_z^{-1}(y_i)$ asked by Algorithm $\mathcal{A}$. Hence, Algorithm $\mathcal{B}$ can obtain θ with non-negligible probability $1/w$ at least from the forged signature.

Table 1 Comparison with the BSS scheme.

		BSS	Prop.
size		$2^k n \log n$	$3n - k + 1$
signing	g_i	$2^k n \log n - k$	$2(n-k)$
	g_i^{-1}	k	$2k$
verification		$2^k n \log n$	$2n$

3.4 Efficiency

In this section, we compare the proposed scheme with the BSS scheme in terms of the size of a signature and the time of procedures. The results are summarized in **Table 1**.

3.4.1 Size of a Signature

We discuss the relationship of the size of a signature, the number of members n, and the number of signers k. The number of data to represent curve C is $n-k+1$. The number of α_i and β_i is $2n$. Accordingly, the essential number of data of a signature is $3n - k + 1$. In the case of the BSS scheme, the essential number of data of a signature is $2^k n \log n$. When $k \geq 2$, we have $3n - k + 1 < 2^k n \log n$. Therefore, our scheme is better than the BSS scheme with respect to the size of a signature. In both of the above evaluations, the public key P_i is ignored because it is public.

3.4.2 Computing Time

We discuss the number of computations of g_i and g_i^{-1}, which is related with the computing time of procedures.

In the signing procedure of the proposed scheme, the number of computations of g_i is $2(n-k)$, and that of g_i^{-1} is $2k$. In the verification procedure of the proposed scheme, the number of computations of g_i is $2n$.

In the signing procedure of the BSS scheme, the number of computations of g_i is $2^k n \log n - k$, and that of g_i^{-1} is k. In the verification procedure of the BSS scheme, the number of computations of g_i is $2^k n \log n$.

Let us compare the signing procedures. Usually, the exponent of the RSA function is small.

e.g., $e = 3$. Suppose that the 1024-bit modulo binary method is used for the exponentiation. Then, the computing time of g_i^{-1} is 768 times as long as that of g_i on average. The computing time of our scheme can be considered as $2(n-k)+768\cdot 2k$, and that of the BSS scheme can be done as $2^k n\log n - k + 768k$.

$$\begin{aligned}&(2^k n\log n - k + 768k) - (2(n-k)+1536k)\\ &= 2^k n\log n - 2n - 767k\\ &\geq 2^k n\log n - 769n\end{aligned}$$

Hence, when $k \geq 10$, the signing procedure of our scheme is more efficient than that of the BSS scheme.

The verification procedure of our scheme is always more efficient than that of the BSS scheme. It should notice that the proposed scheme does not depend on the number of signers.

In the proposed scheme, the computation for obtaining coefficients of the curve is required. However, when n is not so large, the dominant factor of the computing time is the computation of g_i^{-1}. In the case of the BSS scheme, the computation for obtaining the complete partitioning system is required, but it is ignored in the above discussion.

4. Modifications

4.1 Improvement on the Efficiency

The proposed scheme uses $\mathrm{GF}(2^b)$ as the common domain where 2^b is much larger than N_i. If the size of the common domain can be reduced, then the efficiency is improved. In this section, we discuss use of a smaller common domain. Specifically, we use $\mathrm{GF}(2^d)$ as the common domain where 2^d is smaller than N_i. For example, when N_i is a 1024-bit number, d is 256.

We define a mapping $E_{i,z}$ from Z_{N_i} to $\mathrm{GF}(2^d)$ for $z \in \{0,1\}^\ell$ and its inverse mapping $E_{i,z}^{-1}$ as follows. Let $\widehat{E}_z$ be a public symmetric cipher $\{0,1\}^d \times \{0,1\}^\ell \to \{0,1\}^d$. Let $\mathrm{LSB}_t(x)$ be the least significant t bits of x and $|N_i|$ the number of bits of N_i. Then, we define, for $x \in Z_{N_i}$,

$$E_{i,z}(x) = \widehat{E}_z(\mathrm{LSB}_d(x)), \tag{3}$$

and for $y \in \{0,1\}^d$,

$$E_{i,z}^{-1}(y) = r_{|N_i|}||r_{|N_i|-1}||\cdots||r_{|N_i|-d}||\widehat{E}_z^{-1}(y), \tag{4}$$

where $||$ is the concatenation operator and r_i is a random bit such that the right side of Eq. (4) is less than N_i where the most significant bit is $r_{|N_i|}$. Since 2^d is smaller than N_i, $E_{i,z}^{-1}$ is a probabilistic function satisfying $w = E_{i,z}(E_{i,z}^{-1}(w))$ for any $w \in \mathrm{GF}(2^d)$.

Then, steps 3, 4, 5 in the signing procedure described in Section 3.2 are changed as follows.

step 3 For $i \in \bar{\mathcal{S}}$, the signers uniformly choose α_i from the domain of f_i at random, and compute $x_i = E_{i,z}(f_i(\alpha_i))$. If $x_i = x_{i'}$ for $i, i' \in \bar{\mathcal{S}}$, then the signers choose α_i again. For $i \in \bar{\mathcal{S}}$, the signers uniformly choose β_i from the domain of f_i at random, and compute $y_i = E_{i,z}(f_i(\beta_i))$.

step 4 The signers determine the lowest-degree curve C that goes through the $n-k+1$ points (x_i, y_i) $(i \in \bar{\mathcal{S}} \cup \{0\})$.

$$C : y = c(x) = \sum_j c_j x^j, \quad c_j \in \mathrm{GF}(2^d)$$

If the degree of C is not equal to $n-k$, then the signers go back to step 3.

step 5 For $i \in \mathcal{S}$, the signers uniformly choose $x_i \in \{0,1\}^d$ at random. If $x_i = x_{i'}$ for $i, i' \in \bar{\mathcal{S}}$, then the signers choose x_i again. Then, the signers compute $y_i = c(x_i)$ for $i \in \mathcal{S}$. Using their secret keys, they compute $\alpha_i = f_i^{-1}(E_{i,z}^{-1}(x_i))$ and $\beta_i = f_i^{-1}(E_{i,z}^{-1}(y_i))$.

The verification procedure described in Section 3.2 are changed as follows.

step 1 The verifier checks that $1 \leq k \leq n-1$. If it does not hold, then the verifier rejects m and k. The verifier checks that C is an $(n-k)$-degree curve over $\mathrm{GF}(2^d)$ and α_i, β_i $(i \in \mathcal{R})$ are elements in Z_{N_i}. If one of them is not satisfied, then the verifier rejects m and k.

step 2 The verifier computes

$$z = h'(m, k, P_1, P_2, \ldots, P_n).$$

For $i \in \mathcal{R} \cup \{0\}$, the verifier computes

$$x_i = \begin{cases} 0 & \text{if } i = 0,\\ E_{i,z}(f_i(\alpha_i)) & \text{if } i \in \mathcal{R}, \end{cases}$$

$$y_i = \begin{cases} h(m, k, P_1, P_2, \ldots, P_n) & \text{if } i = 0,\\ E_{i,z}(f_i(\beta_i)) & \text{if } i \in \mathcal{R}. \end{cases}$$

step 3 The verifier checks that all the $(n+1)$ points lie on C over $\mathrm{GF}(2^d)$. If it does not hold, then the verifier rejects m and k.

The identity of the signers is unconditionally protected with the above scheme. For $i \in \bar{\mathcal{S}}$, the distribution of x_i is uniform on $\{0,1\}^d$ because the distribution of $\mathrm{LSB}_d(f(\alpha_i))$ can be considered as uniform. For $i \in \mathcal{S}$, the distribu-

tion of x_i is also uniform on $\{0,1\}^d$ due to the signing procedure. For $i \in \bar{\mathcal{S}}$, the distributions of α_i and β_i are uniform on N_i due to the signing procedure. For $i \in \mathcal{S}$, the distribution of α_i is uniform on N_i because the distribution of the right side of Eq. (4) is uniform on N_i. The distribution of β_i is also uniform on N_i because α_i and β_i for $i \in \bar{\mathcal{S}}$ are uniformly distributed and α_i for $i \in \mathcal{S}$ is uniformly done.

On the other hand, the difficulty of the forgery is an open problem. Since the signers (the adversary) can choose r_i, we cannot prove the difficulty of the forgery in the manner similar to Section 3.3.2. In addition, selecting of r_i might enable the adversary to compute f_i^{-1} easily.

However, we conjecture that the above scheme is secure from the following consideration. In order to forge a signature, the adversary has to place $n+1$ points on the $(n-k)$-degree curve. Due to $E_{i,z}$, the probability that $n+1$ random points lie on the $(n-k)$-degree curve is 2^{-kd}, which is negligible. Hence, it seems that the adversary has no choice but to place k points on the $(n-k)$-degree curve that is uniquely determined by the $n-k+1$ points. In order to place the k points on the $(n-k)$-degree curve, the adversary has to compute f_i^{-1}. The computation of f_i^{-1} seems to be difficult even if r_i can be selected.

4.2 Application to ElGamal's Signature Scheme

We show a (k,n) threshold ring signature scheme based on ElGamal's signature scheme[5], which is not based on the trapdoor one-way permutation. Each ring member A_i has an ElGamal public key P_i that specifies verification keys u_i, v_i, p_i and a verification function

$$v_i^m \equiv u_i^t t^s \pmod{p_i},$$

where (t,s) is a signature of m. Usually the hashed value of m is signed, but the hash function is not applied to m in this section.

Similar to Section 4.1, we use a mapping $E_{i,z}$ from Z_{p_i} to $\mathrm{GF}(2^d)$ for $z \in \{0,1\}^\ell$ and its inverse mapping $E_{i,z}^{-1}$. Here, 2^d is smaller than any of the p_i's (e.g., $d=256$). In addition, h is a public collision-resistant hash function from $\{0,1\}^*$ to $\mathrm{GF}(2^d)$. The finite field $\mathrm{GF}(2^d)$ is used as the common domain.

The signing procedure is given as follows.

step 1 The k signers select $n-k$ non-signers arbitrarily. Suppose that $\mathcal{R} = \{1,2,\ldots,n\}$.

step 2 The signers define (x_0, y_0) and z as

$$x_0 = 0,$$
$$y_0 = h(m,k,P_1,P_2,\ldots,P_n),$$
$$z = h'(m,k,P_1,P_2,\ldots,P_n).$$

step 3 For $i \in \bar{\mathcal{S}}$, the signers find $\alpha_i, \beta_i, \gamma_i$ satisfying

$$v_i^{\beta_i} \equiv u_i^{\alpha_i} \alpha_i^{\gamma_i} \pmod{p_i} \qquad (5)$$

by using the existential forging method[5]. That is, they randomly choose η_i and λ_i satisfying $\gcd(\lambda_i, p_i - 1) = 1$ from Z_{p_i-1}. and compute

$$\alpha_i = v_i^{\eta_i} u_i^{\lambda_i} \bmod p_i$$
$$\beta_i = -\frac{\alpha_i \eta_i}{\lambda_i} \bmod (p_i - 1)$$
$$\gamma_i = -\frac{\alpha_i}{\lambda_i} \bmod (p_i - 1).$$

They compute $x_i = E_{i,z}(\alpha_i)$ and $y_i = E_{i,z}(\beta_i)$. If $x_i = x_{i'}$ for $i, i' \in \bar{\mathcal{S}}$, then the signers choose α_i again.

step 4 The signers determine the lowest-degree curve C that goes through the points (x_i, y_i) $(i \in \bar{\mathcal{S}} \cup \{0\})$.

$$C : y = c(x) = \sum_j c_j x^j, \quad c_j \in \mathrm{GF}(2^d)$$

If the degree of C is not equal to $n-k$, then the signers go back to step 3.

step 5 For $i \in \mathcal{S}$, the signers choose $r_i \in Z_{p_i-1}$ satisfying $\gcd(r_i, p_i - 1) = 1$ at random, and compute $\alpha_i = v_i^{r_i} \bmod p_i$ and $x_i = E_{i,z}(\alpha_i)$. If $x_i = x_{i'}$ for $i, i' \in \mathcal{S}$, then the signers choose r_i again. After the signers computed $y_i = c(x_i)$, the signers obtain $\beta_i = E_{i,z}^{-1}(y_i)$. Then, the signers compute γ_i satisfying the following equation by using the secret keys.

$$v_i^{\beta_i} \equiv u_i^{\alpha_i} \alpha_i^{\gamma_i} \pmod{p_i}$$

step 6 Finally, the (k,n) threshold ring signature of m is defined as

$$\sigma = \{C, k, (P_i, \alpha_i, \beta_i, \gamma_i)\ (i \in \mathcal{R})\}.$$

The verification procedure is given as follows.

step 1 The verifier checks that $1 \le k \le n-1$. If it does not hold, then the verifier rejects m and k. The verifier checks that C is an $(n-k)$-degree curve over $\mathrm{GF}(2^d)$ and $\alpha_i, \beta_i, \gamma_i$ $(i \in \mathcal{R})$ are elements in Z_{p_i}, Z_{p_i-1} and Z_{p_i-1}, respectively. If one of them does not hold, then the verifier rejects m and k.

step 2 The verifier checks that the following equation holds for any $i \in \mathcal{R}$.

$$v_i^{\beta_i} \equiv u_i^{\alpha_i} \alpha_i^{\gamma_i} \pmod{p_i}$$

step 3 The verifier computes
$z = h'(m, k, P_1, P_2, \ldots, P_n)$.
For $i \in \mathcal{R} \cup \{0\}$, the verifier computes

$$x_i = \begin{cases} 0 & \text{if } i = 0, \\ E_{i,z}(\alpha_i) & \text{if } i \in \mathcal{R}, \end{cases}$$

$$y_i = \begin{cases} h(m, k, P_1, P_2, \ldots, P_n) & \text{if } i = 0, \\ E_{i,z}(\beta_i) & \text{if } i \in \mathcal{R}. \end{cases}$$

The verifier checks that $y_i = c(x_i)$ over $\mathrm{GF}(2^d)$ for all $i \in \mathcal{R} \cup \{0\}$. If it holds for $i \in \mathcal{R} \cup \{0\}$, then the verifier accepts m and k. Otherwise the verifier rejects them.

The identity of the signers is unconditionally protected with the above scheme. The distributions of α_i, β_i, and γ_i are same for the signers and the non-signers; α_i is uniformly distributed on the set of generators of $\mathrm{GF}(p_i)$, β_i is uniformly distributed on Z_{p_i-1}, and γ_i is uniquely determined by α_i and β_i. Notice that the existential forging method in step 3 of the signing procedure requires neither the secret key nor a previous valid signature[5].

On the other hand, the difficulty of the forgery is an open problem. Since $E_{i,z}^{-1}$ is given by Eq. (4), selecting of r_i might enable the adversary to find α_i, β_i, and γ_i such that the points lies on the $(n-k)$-degree curve. However, we conjecture that the above scheme is secure from the consideration similar to Section 4.1.

We discussed a ring of RSA-based members and a ring of ElGamal-based members separately. However, when the above small domain is used, the difference is how to compute x_i, y_i. When (x_i, y_i) are once decided, the curve C is determined from the points, and does not depend on the individual signature schemes. The change of computation of (x_i, y_i) corresponding to ring members enables RSA-based members and ElGamal-based members to join the same ring.

5. Concluding Remarks

In this paper, we have shown the provably secure (k, n) threshold ring signature scheme. While the RST scheme and the BSS scheme form the ring of individual signatures, our scheme forms the curve of them. Our scheme is more efficient than the BSS as follows. The size of a signature of our scheme is smaller than that of the BSS scheme. The computing time of the signing procedure of our scheme is shorter than that of the BSS scheme when $k \geq 10$. The computing time of the verification procedure of our scheme is shorter than that of the BSS scheme.

We have also discussed the reduction of the common domain, and have shown the (k, n) threshold ring signature scheme with the small domain. Using the small domain, we have constructed the (k, n) threshold ring signature scheme based on ElGamal's signature scheme. This scheme utilizes the fact that ElGamal's signature scheme is existentially forgeable. We note that the security of the schemes with the small domain is not proved; it is an open problem.

Acknowledgments The authors thank to the anonymous reviewers for their useful comments. This research was partially supported by the Ministry of Education, Culture, Sports Science and Technology, Grant-in-Aid for Encouragement of Young Scientists (B).

References

1) Bresson, E., Stern, J. and Szydlo, M.: Threshold Ring Signatures and Applications to Ad-hoc Groups, *Advances in Cryptology — CRYPTO 2002, Lecture Notes in Computer Science*, Vol.2442, pp.465–480 (2002).
2) Camenisch, J.: Efficient and Generalized Group Signatures, *Advances in Cryptology — EUROCRYPT'97, Lecture Notes in Computer Science*, Vol.1233, pp.465–479 (1997).
3) Chaum, D. and van Heyst, E.: Group Signatures, *Advances in Cryptology — EUROCRYPT'92, Lecture Notes in Computer Science*, Vol.658, pp.366–377 (1993).
4) Diffie, W. and Hellman, M.E.: New Directions in Cryptography, *IEEE Trans. Inf. Theory*, Vol.IT-22, No.6, pp.644–654 (1976).
5) ElGamal, T.: A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms, *IEEE Trans. Inf. Theory*, Vol.IT-31, No.4, pp.469–472 (1985).
6) Naor, M.: Deniable Ring Authentication, *Advances in Cryptology — CRYPTO 2002, Lecture Notes in Computer Science*, Vol.2443, pp.481–498 (2002).
7) Ohkubo, M., Abe, M., Suzuki, K. and Tsujii, S.: Short 1-out-n proof, *Proc. 2002 Symposium on Cryptography and Information Security*, Vol.I, pp.189-193 (2002).
8) Rivest, R.L., Shamir, A. and Adleman, L.: A Method for Obtaining Digital Signatures and Public-Key Cryptosystems, *Comm. ACM*, Vol.21, pp.120–126 (1978).
9) Rivest, R.L., Shamir, A. and Tauman, Y.: How to Leak a Secret, *Advances in Cryptology*

— *ASIACRYPT 2001, Lecture Notes in Computer Science*, Vol.2248, pp.552–565 (2001).

(Received November 29, 2002)
(Accepted June 3, 2003)

Hidenori Kuwakado received the B.E., M.E. and D.E. degrees from Kobe University in 1990, 1992, and 1999 respectively. He worked for Nippon Telegraph and Telephone Corporation from 1992 to 1996. From 1996 to 2002 he was a Research Associate in the Faculty of Engineering, Kobe University. Since 2002, he has been an Associate Professor in the Faculty of Engineering, Kobe University. His research interests are in cryptography and information security.

Hatsukazu Tanaka was born in Hyogo, Japan, on September 30, 1941. He received the B.E. degree from Kobe University, Kobe, Japan in 1964, the M.E. degree in 1966, and the D.E. degree in 1969, both from Osaka University, Osaka, Japan. He was appointed as a Research Associate in the Faculty of Engineering, University of Osaka Prefecture in 1969. From 1972 through 1987 he was an Associate Professor in the Department of Electrical Engineering, Kobe University. Since 1988 he has been a Professor in the Department of Electrical and Electronics Engineering, Kobe University. From 1980 through 1981 he was a member of the Communication Group of the University of Toronto, Toronto, Ontario, Canada, as a Visiting Scientist. His main work is on the basic theory of Information Engineering such as Information Theory, Coding Theory, Cryptography and Information Security, Image Processing, etc. Dr. Tanaka is a Fellow member of IEEE, a Fellow memeber of IEICE, and a member of IACR.