Generalized Ring Signatures

Jian Ren, *Member*, *IEEE*, and Lein Harn

Abstract—The concept of ring signature was first introduced by Rivest et al. in 2001. In a ring signature, instead of revealing the actual identity of the message signer, it specifies a set of possible signers. The verifier can be convinced that the signature was indeed generated by one of the ring members; however, the verifier is unable to tell which member actually produced the signature. In this paper, we first propose a generalized ring signature scheme based on the original ElGamal signature scheme. The proposed ring signature can achieve unconditional signer ambiguity and is secure against adaptive chosen-message attack in the random oracle model. Then, based on the generalized ring signature scheme, a generalized multisigner ring signature scheme is introduced to increase the level of confidence or enforce cross-organizational joint message signing. Comparing to ring signatures based on RSA algorithm, the proposed generalized ring signature scheme has three advantages: 1) all ring members can share the same prime number p and all operations can be performed in the same domain; 2) by combining with multisignatures, we can develop the generalized multisigner ring signature schemes to enforce cross-organizational involvement in message leaking. It may result in a higher level of confidence or broader coverage on the message source; 3) the proposed ring signature is convertible. It enables the actual message signer to prove to a verifier that only she is capable of generating the ring signature.

Index Terms—Generalized ring signature, anonymity, unconditional secure, unforgeability, random oracle secure.

1 Introduction

The concept of ring signature was first introduced by Rivest et al. in 2001 [1] to provide anonymity for the message signer. In a ring signature scheme, the message signer, say Alice, selects a set of ring members including herself as the possible message signers. The actual message signer can generate a ring signature on her own using only her private key and the others' public keys, without the other ring members' assistance or even awareness. However, the generated ring signature can convince any verifier that the message was indeed signed by one of the ring members while the real signer's identity is totally anonymous to the verifier.

The idea behind ring signature schemes is similar to that of group signatures [2], [3], [4] but with some variations. First of all, unlike a group signature, a ring signature scheme does not require a group manager to administrate the set of ring members. The actual message signer has the freedom to select all the ring members and sign whatever messages she like. Second, in a group signature scheme, the group manager can recover the real identity of the actual message signer. In fact, a group signature only looks indistinguishable to the verifier but not to the group manager. The group manager can even revoke the anonymity of misbehaving signers.

Since the introduction of ring signature, several ring signature schemes have been proposed. In [5], a ring signature based on Schnorr signature scheme [6] is proposed. Recently, ring signature schemes based on bilinear pairing [7], [8], [9], [10] and identity-based ring signature schemes [9], [11] are introduced. Some variations of ring signature schemes are also presented in literature [10], [12], [13], [14], [15]. Similar to group signatures, convertibility has also been defined for ring signature in the literature [16]. By definition, convertibility enables the actual message signer to provide nonrepudiation evidence to a verifier for the originality of the signature at times of her choice. However, we point out in this paper that the convertibility algorithm proposed in [16] cannot produce nonrepudiation evidence of the actual message signer.

In this paper, a ring signature scheme based on the original ElGamal signature scheme is first proposed. We call it a *generalized ring signature*. The generalized ring signature scheme is secure against adaptive chosen-message attack [17]. This means that the proposed generalized ring signature scheme is secure even if the adversary is allowed not only to get ring signatures for whatever messages she like but also to request ring signatures of messages that depend additionally on previously obtained signatures. Comparing to ring signature schemes based on RSA, where each ring member uses a different modulus that has to be expanded to a common domain, in the generalized ring signature scheme, all ring members can share the same prime number and all operations can be performed in the same domain. Moreover, the generalized ring signature scheme is convertible. The actual message signer can always prove to a verifier for the originality of the ring message at times of her choice. Next, we define a generalized ring signature scheme with multiple signers. We call it the *generalized multisigner ring signature*. Though the concept of the generalized multisigner ring signature scheme is similar to that of a threshold ring signature scheme [18], [19], [20], [21], the generalized multisigner ring signature maintains the ring structure defined in [1]. The difference also includes that in a threshold ring signature, all the n possible signers are equally possible in generating the ring signature, while only t are the actual signers. While for the proposed generalized multisigner ring

- *J. Ren is with the Department of Electrical and Computer Engineering, 2120 Engineering Building, Michigan State University, East Lansing, MI 48864-1226. E-mail: renjian@egr.msu.edu.*
- *L. Harn is with the Department of Computer and Electrical Engineering, University of Missouri, Kansas City, MO 64110-2499. E-mail: harnl@umkc.edu.*

Manuscript received 3 Oct. 2006; revised 18 Sept. 2007; accepted 8 Jan. 2008; published online 17 Mar. 2008.
For information on obtaining reprints of this article, please send e-mail to: tdsc@computer.org, and reference IEEECS Log Number TDSC-0139-1006.
Digital Object Identifier no. 10.1109/TDSC.2008.22.

signature, each ring member can contain a group of users. The diversity of the ring members can enforce cross-organizational message signing, while the actual number of signers can be hidden from the verifiers.

Our contributions in this paper are primarily twofold. First, a generalized ring signature scheme based on the original ElGamal signature scheme is proposed. Second, a generalized multisigner ring signature scheme that combines multisignature and the generalized ring signature is proposed to increase the level of confidence or enforce cross-organizational joint message signing. Both the proposed generalized ring signature scheme and the generalized multisigner ring signature scheme can achieve unconditional signer ambiguity and are secure against adaptive chosen-message attacks. Moreover, the proposed schemes have two advantages over the ring signature schemes based on RSA: 1) the extension operation can be avoided since all ring members can share the same prime modulus p and all operations can be performed in the same domain and 2) the generalized ring signature scheme is convertible. The actual message signer can prove to a verifier that only she is capable of generating the ring signature without requiring any extra effort.

This paper is organized as follows: In Section 2, we give our notation and the preliminaries. In Section 3, we propose a generalized ring signature based on the ElGamal signature scheme. We also prove that the proposed generalized ring signature scheme is secure against adaptive chosen-message attack. In Section 4, we further define a generalized multisigner ring signature scheme. Convertibility discussion of RSA-based ring signature scheme and the proposed ring signature schemes are presented in Section 5 along with security analysis of the proposed schemes. We conclude in Section 6.

2 Preliminaries

2.1 Notation

In [1], the concept of ring signature was first proposed. Suppose that Alice wishes to generate a ring signature of a message m for a ring of n individuals $A_1, A_2, \ldots, A_n$, where the signer Alice is $A_s, 1 \le s \le n$. Denote $\mathcal{S} = \{A_1, A_2, \ldots, A_n\}$. Each $A_i \in \mathcal{S}$ is called a *ring member*. The public key of A_i is P_i and the corresponding private key is S_i. In this paper, we will not distinguish between the ring member and its public key. Therefore, $\mathcal{S}$ will also be used to denote the set of public keys of all ring members.

A *ring signature* scheme consists of the following two algorithms:

- ring-sign $(m, \mathcal{S})$. Given a message m and the set of ring members $\mathcal{S} = \{P_1, P_2, \ldots, P_n\}$, the actual signer A_s can produce a ring signature σ using $\mathcal{S}$ and her own private key S_s.
- ring-verify (m, σ). Given a message m and a ring signature σ, which includes $\mathcal{S} = \{P_1, P_2, \ldots, P_n\}$, a verifier can determine whether (m, σ) is a valid ring signature generated by one of the ring members.

There are two security requirements for ring signature schemes, which include

1. **Signer ambiguity.** The probability that a verifier can successfully determine the real signer of a ring signature is exactly $1/n$, where n is the total number of ring members.
2. **Unforgeability.** The advantage that a non-ring-member can successfully forge a ring signature is negligible.

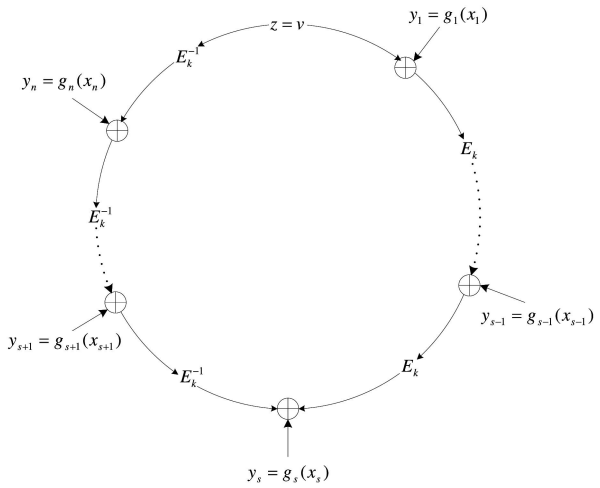


Fig. 1. Ring signatures.

Combining functions. A combining function $C_{k,v}(y_1, y_2, \ldots, y_n)$ takes as input a key k, an initialization value v, and a list of arbitrary values $y_1 = g_1(x_1), \ldots, y_n = g_n(x_n) \in \{0,1\}^b$, where $g_1, \ldots, g_n$ are trapdoor functions. It outputs a value $z \in \{0,1\}^b$, such that for any given k, v, s, $1 \le s \le n$, and any fixed values of all the other inputs y_i, $i \ne s$, the function $C_{k,v}$ is a one-to-one mapping from y_s to the output z. Moreover, this mapping is efficiently solvable. However, it should be infeasible to solve the verification equation for $x_1, \ldots, x_n$ without knowing any of the private keys and inverting any of the trapdoor functions $g_1, \ldots, g_n$.

In [1], a combining function is proposed as follows:

$$\begin{aligned} z &= C_{k,v}(g_1(x_1), \ldots, g_n(x_n)) \\ &= E_k(g_n(x_n) \oplus E_k(\cdots \oplus E_k(g_1(x_1) \oplus v))). \end{aligned} \tag{1}$$

Equivalently, we have

$$\begin{aligned} y_s = {} & E_k(g_{s-1}(x_{s-1}) \oplus \cdots \oplus E_k(g_1(x_1) \oplus)) \\ & \oplus E_k^{-1}\big(g_{s+1}(x_{s+1}) \oplus \cdots \oplus E_k^{-1}\big(g_n(x_n) \oplus E_k^{-1}(z)\big)\big). \end{aligned} \tag{2}$$

2.2 Ring Signatures

The ring signature proposed by Rivest et al. [1] is based on the RSA signature scheme [22]. We call it *RSA-based ring signature*. The main idea of the RSA-based ring signature is illustrated in Fig. 1 with the combining function defined in (1) and (2) above.

In the RSA-based ring signature scheme, each ring member A_i has an RSA public key $P_i = (n_i, e_i)$, which specifies the trapdoor one-way permutation f_i over $\mathbb{Z}_{n_i}$:

$$f_i(x_i) = x_i^{e_i} \bmod n_i.$$

It is assumed that only A_i knows how to compute the inverse permutation f_i^{-1} efficiently.

One of the problem that RSA algorithm faces is that each ring member has different modulus, which makes it awkward to combine the individual signatures. To solve

this problem, all the trapdoor permutations are extended to a common domain $\{0,1\}^b$, where 2^b is some power of two, which is larger than all the moduli n_i's. The extended trapdoor permutation g_i over $\{0,1\}^b$ is defined in the following way. For any b-bit input m_i, let $m_i = q_i n_i + r_i$, where q_i and r_i are nonnegative integers, $0 \le r_i \le n_i$. Then

$$g_i(m_i) = \begin{cases} q_i n_i + f_i(r_i), & \text{if } (q_i+1)n_i \le 2^b, \\ m_i, & \text{else.} \end{cases}$$

It is assumed that there is the existence of a publicly defined ideal symmetric encryption algorithm E such that for any k of length l, the function E_k is a permutation over b-bit strings. It is also assumed that there is the existence of a publicly defined collision-resistance hash function h that maps arbitrary inputs to strings of length l, which are used as keys for E.

In this section, we will describe the RSA-based ring signature scheme proposed in [1], which contains the two algorithms below:

ring-sign $(m, \mathcal{S})$. Suppose that Alice wishes to sign a message m with a ring signature for the ring of n individuals $A_1, A_2, \ldots, A_n$, where Alice is A_s for some s, $1 \le s \le n$. Given the message m to be signed, A_s's private key $S_s = (d_s, n_s)$, and the sequence of public keys $P_1, P_2, \ldots, P_n$ of all the ring members, A_s computes a ring signature as follows:

1. **Choose a key.** The signer A_s first computes the symmetric key k as follows:
$$k = h(m).$$
2. **Pick a random glue value.** The signer picks an initialization value $v \in \{0,1\}^b$ uniformly at random.
3. **Pick random x_i's.** A_s picks random x_i for all the other ring members $1 \le i \le n$, $i \ne s$ uniformly and independently from $\{0,1\}^b$, and computes
$$y_i = g_i(x_i).$$
4. **Solve for y_s.** A_s solves the following ring equation for y_s:
$$C_{k,v}(y_1, y_2, \ldots, y_n) = v.$$
Equivalently, we can solve y_s as follows:
$$y_s = E_k(y_{s-1} \oplus E_k(\cdots \oplus E_k(y_1 \oplus v))) \oplus E_k^{-1}\big(y_{s+1} \oplus E_k\big(\cdots \oplus E_k^{-1}\big(y_n \oplus E_k^{-1}(v)\big)\big)\big).$$
5. **Invert y_s using A_s's trapdoor permutation.** A_s uses her knowledge of the trapdoor to invert g_s on y_s to obtain x_s:
$$x_s = g_s^{-1}(y_s).$$
6. **Output the ring signature.** The signature on the message m is defined to be
$$\sigma = (\mathcal{S}; v; x_1, x_2, \ldots, x_n). \quad (3)$$

ring-verify $(m; \sigma; \mathcal{S})$. A verifier can check an alleged signature on message m as follows:

1. **Apply the trapdoor permutations.** For $i = 1, 2, \ldots, n$, the verifier computes
$$y_i = g_i(x_i).$$
2. **Obtain k.** The verifier hashes the message m:
$$k = h(m).$$
3. **Verify the ring equation.** The verifier checks that the y_i's satisfy
$$C_{k,v}(y_1, y_2, \ldots, y_n) = v.$$

If the ring equation is satisfied, the verifier Accepts the ring signature as valid. Otherwise, the verifier Rejects.

3 Proposed Generalized Ring Signature

In this section, we introduce a *generalized ring signature* scheme based on the original ElGamal signature scheme.

In an ElGamal signature scheme, a large prime number p and a primitive element g in $\mathbb{Z}_p$ are assumed to be made publicly known. The signer can select a random $d \in \mathbb{Z}_{p-1}$ as her private key. Then, the public key is computed from $e = g^d \bmod p$.

Let m be the message to be signed. The signer randomly selects a one-time secret $l \in \mathbb{Z}_{p-1}$ and computes $\alpha = g^l \bmod p$. Then she computes $\beta = (m - d\alpha)l^{-1} \bmod p - 1$. The signature for message m is defined as the pair (α, β). The signature can be verified if $g^m = e^\alpha \alpha^\beta \bmod p$ is true.

The construction of a ring signature requires existential forgery. According to [17] and [23], the ElGamal signature scheme is existentially forgeable with a generic message attack. In fact, there are two well-known levels of forgeries: one-parameter forgery and two-parameter forgery.

For one-parameter forgery, select $c \in \mathbb{Z}_{p-1}$ arbitrarily, if we let $\alpha = g^c e \bmod p$ and $\beta = -\alpha \bmod p - 1$, then it is easy to see that (α, β) is a valid signature for the message $m = c\beta \bmod p - 1$. However, this forgery is easily detectable since $\alpha + \beta = 0 \bmod p - 1$, we will not use this method of forgery.

For the second level of forgeries [23], the actual message signer selects $a_i \in \mathbb{Z}_{p-1}$ and $b_i \in \mathbb{Z}_{p-1}^*$ arbitrarily for each $i = 1, 2, \ldots, n$, $i \ne s$. Define

$$\alpha_i = g^{a_i} e_i^{b_i} \bmod p, \quad (4)$$

$$\beta_i = -\alpha_i b_i^{-1} \bmod p - 1, \quad (5)$$

$$m_i = a_i \beta_i \bmod p - 1, \quad (6)$$

then it can be shown that (α_i, β_i) is a valid signature of message m_i.

Define

$$g_i(a_i, b_i) = (m_i, \alpha_i, \beta_i),$$

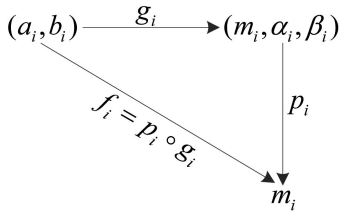


Fig. 2. Existential forgery of the ElGamal signature scheme.

then the inverse g_i is easy to compute. However, if we compose g_i with the project mapping p_i, where

$$p_i(m_i, \alpha_i, \beta_i) = m_i,$$

then the inverse of $f_i = p_i \circ g_i$ is computationally difficult. In fact, the inverse of f_i, shown in Fig. 2, is a one-way trapdoor function over $\mathbb{Z}_{p-1} \times \mathbb{Z}_{p-1}^* \mapsto \mathbb{Z}_{p-1}$.

In the proposed ring signature scheme, the same combining function described in (1) can still be used. The assumption for the existence of a collision-resistance hash function h remains the same. However, instead of a publicly defined ideal symmetric encryption algorithm E, we can reduce it to a hash function h, thus we can replace $E_{h(m)}(x)$ with $h(m, x)$ as described in [18].

Comparing to the original RSA-based ring signature scheme, for a generalized ring signature scheme, all ring members can share the same prime number p and all operations can be performed in the same domain $\mathbb{Z}_p$. Therefore, no expansion is necessary. We can assume that the private key of the ith ring member $A_i \in \mathcal{S}$ is $d_i \in \mathbb{Z}_p^*$, $i = 1, 2, \ldots, n$, and the corresponding public key of the ring member is given by $e_i = g^{d_i} \bmod p$.

We will now describe the ring-sign and ring-verify procedures, shown in Fig. 3, below.

ring-sign $(m, \mathcal{S}, d_s, s)$. Given a message m to be signed, her private key d_s, and the sequence of public keys $e_1, e_2, \ldots, e_n$ of all the ring members, the signer computes a ring signature as follows:

1. **Pick a random glue value.** The signer picks an initialization value v uniformly at random from $\mathbb{Z}_p$.
2. **Create a signature forgery** (α_i, β_i) **for message** m_i. The purpose of this step is to forge a signature for some message m_i for each of the $n-1$ nonsigner ring members using the two-parameter forgery. To achieve this goal, the actual signer selects $a_i \in \mathbb{Z}_{p-1}$ and $b_i \in \mathbb{Z}_{p-1}^*$ arbitrarily for each $i = 1, 2, \ldots, n$, $i \neq s$, then (α_i, β_i) can be derived from
$$g_i(a_i, b_i) = (m_i, \alpha_i, \beta_i).$$
3. **Solve for** m_{i_s}. Suppose that the index for the signer is i_s. Let
$$\begin{aligned} v_{i_s+1} &= h(m, v), \\ v_{i_s+2} &= h(m, v_{i_s+1} \oplus m_{i_s+1}), \\ &\vdots \\ v_{i_s-1} &= h(m, v_{i_s-2} \oplus m_{i_s-2}), \\ v_{i_s} &= h(m, v_{i_s-1} \oplus m_{i_s-1}), \end{aligned}$$

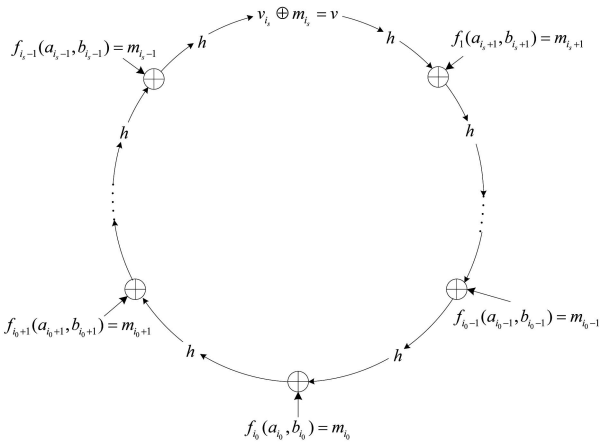


Fig. 3. Proposed ring signatures based on the ElGamal signature scheme.

where the indices are all in $\mathbb{Z}_n$. Therefore, $v_{i_s+n} = v_{i_s}$. To glue the ring, let

$$v_{i_s} \oplus m_{i_s} = v.$$

Equivalently,

$$m_{i_s} = v \oplus v_{i_s}.$$

4. **Sign** m_{i_s} **using the signer's trapdoor permutation.** The signer uses her knowledge of the trapdoor information to sign the message m_{i_s} and obtain the signature (α_s, β_s). To do so, the actual signer A_s first selects a random l uniformly from $\mathbb{Z}_p^*$, such that $\gcd(l, p-1) = 1$. The signature for message m_{i_s} is $(\alpha_i, \beta_i) = (g^l \bmod p, (m_{i_s} - d_i \alpha_i) l^{-1} \bmod p - 1)$.
5. **Output the ring signature.** The signature on the message m is defined as
$$\boldsymbol{\sigma} = (\mathcal{S}; i_0, v_{i_0}; m_1, \ldots, m_n; \alpha_1, \beta_1, \ldots, \alpha_n, \beta_n),$$
where i_0 is randomly selected from $\mathbb{Z}_n$.

ring-verify $(m; \boldsymbol{\sigma}; \mathcal{S})$. A verifier can verify an alleged signature

$$\boldsymbol{\sigma} = (\mathcal{S}; i_0, v_{i_0}; m_1, \ldots, m_n; \alpha_1, \beta_1, \ldots, \alpha_n, \beta_n)$$

on the message m as follows:

1. **Verify the trapdoor permutations.** For $i = 1, 2, \ldots, n$, the verifier checks the following equation:
$$g^{m_i} = e_i^{\alpha_i} \alpha_i^{\beta_i} \bmod p.$$
If it is satisfied, the verifier continues the rest of the verification. Otherwise, the verifier Rejects.
2. **Verify the ring equation.** The verifier checks that the m_i's satisfy the fundamental equation starting from i_0 with value v_{i_0}
$$v_{i_0} = h\Big(m, m_{i_0+n-1} \oplus h\big(m, m_{i_0+n-2} \oplus h(m, \cdots \oplus h(m, m_{i_0} \oplus v) \cdots)\big)\Big). \quad (7)$$

If the ring equation is satisfied, the verifier Accepts the signature as valid. Otherwise, the verifier Rejects.

3.1 Security Analysis

Similar to [1] and [18], the identity of the signer is unconditionally protected with the proposed ring signature scheme. This is because for each k and v, regardless of the signer's identity, the ring signature has exactly p^{n-1} solutions according to (4), (5), and (6), and all of them can be chosen by the signature generation procedure with equal probability without depending on any complexity-theoretic assumptions or on the randomness of the oracle.

The soundness of the ring signature scheme is computational since the ring signature cannot be stronger than the individual signature scheme used by the possible signer.

Theorem 1. *The generalized ring signature scheme based on ElGamal signature scheme is secure against adaptive chosen-message attack in the random oracle model.*

Proof. To prove this theorem, we only need to prove that in the random oracle model, any forgery algorithm $\mathcal{A}$ with nonnegligible probability of a new ring signature for m' by analyzing polynomially many ring signatures for other chosen message $m \neq m'$ can be turned into an algorithm $\mathcal{B}$, which inverts one of the trapdoor one-way functions f_i on random inputs with nonnegligible probability.

Assume there exists a forging algorithm $\mathcal{A}$ with nonnegligible probability in creating a forgery, where the algorithm accepts the public keys and is given oracle access to a ring signing oracle. In producing a valid ring signature that was not presented to the signing oracle with a nonnegligible probability, algorithm $\mathcal{A}$ can work adaptively, querying the oracle at arguments that may depend on previous answers.

Note that the mapping $f_i : \mathbb{Z}_{p-1} \times \mathbb{Z}_{p-1}^* \mapsto \mathbb{Z}_{p-1}$ defined by $p_i \circ g_i$ is a surjection in Fig. 2, which means that all signatures can be generated this way. It is computationally easy to compute the function g_i and its inverse; however, the computation of the inverse of f_i is computationally infeasible. That is the computation of a_i, b_i such that $(a_i, b_i) = f_i^{-1}(m_i)$ for $i \in \{1, 2, \ldots, n\}$ is computationally infeasible.

Algorithm $\mathcal{B}$ uses algorithm $\mathcal{A}$ as a black box on inputs $f_1, f_2, \ldots, f_n$ and a random m, in order to find a value $(a_i, b_i) = f_i^{-1}(m)$ for some $i \in \{1, 2, \ldots, n\}$. $\mathcal{A}$ must query the oracle h with the message that it is forging for a signature. Assume the probability that $\mathcal{A}$ forges the jth signature that it sends to the oracle h is nonnegligible. We denote this message by m'. Algorithm $\mathcal{B}$ begins by guessing randomly this index j, the probability in guessing the correct value should be nonnegligible.

Algorithm $\mathcal{B}$ simulates $\mathcal{A}$'s oracle in the following way. When $\mathcal{A}$ makes a query to oracle h, the query is answered by a uniformly chosen value; however, if the same value is being queried twice, the answer should be the same. Algorithm $\mathcal{B}$ simulates the ring signature oracle by providing a random vector $(i_0, v_{i_0}; m_1, m_2, \ldots, m_n; \alpha_1, \beta_1, \alpha_2, \beta_2, \ldots, \alpha_n, \beta_n)$ as a ring signature to any query m. It then adjusts the random answers to queries to support the correctness of the ring equation (7) for these messages. To do so, $\mathcal{B}$ choose randomly $n-1$ values $z_1, z_2, \ldots, z_{n-1}$ and set $h(m, v + f_1(a_1, b_1)) = z_1$ and $h(m, z_i + f_{i+1}(a_{i+1}, b_{i+1})) = z_{i+1}$, such that $z_n = v$. Note that $\mathcal{A}$ cannot ask oracle queries that will limit $\mathcal{B}$'s freedom of choice, before providing m to the signing oracle, since all the values v, $z_1, \ldots, z_n$ are chosen randomly by $\mathcal{B}$ and cannot be guessed in advance by $\mathcal{A}$.

In order to simulate the oracle h, algorithm $\mathcal{B}$ checks whether $k = h(m')$. If $k \neq h(m')$ (or if $\mathcal{A}$ has not yet queried its jth query to the oracle h), then each query to h is answered randomly, unless the value of this query has already been determined by $\mathcal{B}$, in which case it is answered with the predetermined value. Note that, so far, the simulated oracles are statistically close to the real oracles, and thus in particular, $\mathcal{A}$ cannot distinguish between the real oracles and the simulated oracles. It remains to simulate the oracle h for $k = h(m')$. The goal of algorithm $\mathcal{B}$ is to compute $(a_i, b_i) = f_i^{-1}(\mathrm{m})$, for some i. The basic idea is to slip the value m as the "gap" between the output and input values of two cyclically consecutive h's along the ring equation of the final forgery, which forces $\mathcal{A}$ to close the gap by providing the corresponding a_i, b_i in the generated signature.

Assume that the ith h was queried before the $(i-1)$th h. $\mathcal{B}$ will guess which query corresponds to the ith h and which query corresponds to the $(i-1)$th h. In fact, if the input to the ith h was z, then $\mathcal{B}$ will set the output of the $(i-1)$th h to be $z + \mathrm{m}$. All other queries are answered randomly.

Note that since m is a random value, the simulated oracle h cannot be distinguished from the real oracles, and therefore, with nonnegligible probability, $\mathcal{A}$ will output a signature $(i_0, v_{i_0}; m_1, m_2, \ldots, m_n; \alpha_1, \beta_1, \alpha_2, \beta_2, \ldots, \alpha_n, \beta_n)$ to a message m'. Moreover, with nonnegligible probability, there exists $i \in \{1, \ldots, n\}$ such that $f_i(a_i, b_i) = \mathrm{m}$ as desired. □

4 Generalized Multisigner Ring Signature

A multisigner ring signature enables more signers to be involved in signing the message. When multiple signers work together in generating a ring signature, it may result in a higher level of confidence or broader coverage on the source of the ring signature. As an example, a multisigner ring signature can be generated to enforce cross-organizational involvement in message leaking.

In this section, we propose a generalized multisigner ring signature scheme. We start with some definitions.

Definition 1 (generalized ring member). *In the original ring signature scheme, each ring member is a single user. When a ring member consists of an arbitrary number of message signers, then the ring member is called a* generalized ring member.

Definition 2 (generalized multisigner ring signature). *For a ring signature, if each ring member is a generalized ring member and the signature of each generalized ring member is a*

multisignature, then this ring signature is called a generalized multisigner ring signature.

The basic idea of the generalized multisigner ring signature is similar to the generalized ring signature. In a generalized ring signature, there are n trapdoor one-way functions, where n is the number of possible ring members. Each ring member corresponds to one trapdoor one-way function with a single private key. In a generalized multisigner ring signature, instead of n possible individual ring members, there are n possible generalized ring members, each generalized ring member is composed of an arbitrary number of signers that generates a multisignature [24]. The generalized ring members do not need to contain the same number of signers. The corresponding multisignature determines the number of actual signers. To achieve efficiency, the multisignature scheme proposed in [24] can be used as the trapdoor one-way function since in this scheme, when a multisignature is generated with the knowledge of multiple private keys, the length and verification time of the multisignature is constant (i.e., not a linear function with respect to the number of signers involved).

The generalized multisigner ring signature also bears some similarity with the threshold ring signature. The major difference between the generalized multisigner ring signature and the threshold ring signature [18] is that for a threshold ring signature, all the n possible signers are equally possible in generating the ring signature, while only t are the actual signers. However, for the proposed generalized multisigner ring signature scheme, this is not necessarily true. Moreover, in a generalized multisigner ring signature, the "deep throat" may include members of cross-organizations such as members from financial organization and also members from management organization. This can be illustrated clearly in Example 1 below.

Example 1. Suppose the possible signers' subset is A, B, C, D, and E. For a two-out-of-five threshold ring signature, any two users could possibly be the message signers. However, for a generalized ring signature, the generalized ring members could be $\{A, B\}$, $\{C, D\}$, $\{C, D, E\}$, $\{A, B, E\}$. Then, the generalized ring signature has either two or three signers since we are unable to determine which generalized ring member is the actual signer.

It is computationally infeasible to determine the actual number of trapdoors involved in creating the ring signature and determine the actual signer of the multisignature if the generalized ring members have different number of signers except that one of the multisignature is actually generated with the knowledge of the trapdoor information. Therefore, the number of actual signers is lower bounded by the generalized ring members with the least number of signers.

The multisignature scheme proposed in [24] is based on a variation of the ElGamal signature of the following form.

Signing. Randomly select $l \in \mathbb{Z}_{p-1}^*$ and compute $\alpha = g^l \bmod p$, $\beta = dm - l\alpha \bmod p - 1$. Then, (α, β) is a signature.

Verification. The signature can be verified if

$$e^m = \alpha^\alpha g^\beta \bmod p$$

is true.

Assume that a subset consists of t signers with public keys $e_1, e_2, \ldots, e_t$ that wish to sign the same message m. The group's public key is defined as

$$e = \prod_{i=1}^{t} e_i \bmod p.$$

The generation of a multisignature can be described as follows:

1. Each signer randomly selects a number $l_i \in \mathbb{Z}_{p-1}$, then computes

   $$\alpha_i = g^{l_i} \bmod p,$$

   and

   $$\alpha = \prod_{i=1}^{t} \alpha_i \bmod p.$$

2. Each signer solves the following equation:

   $$\beta_i = d_i m - l_i \alpha \bmod p - 1,$$

   where d_i is his private key, and l_i is a random number that he selected from $\mathbb{Z}_{p-1}$. Define

   $$\beta = \sum_{i=1}^{t} \beta_i \bmod p - 1.$$

The multisignature (α, β) for message m can be verified if

$$e^m = g^\alpha \alpha^\beta \bmod p$$

is true.

Similar to the original ElGamal signature scheme, this variation of ElGamal signature is also existentially forgeable with two-parameter forgery: the actual signer selects $a \in \mathbb{Z}_{p-1}$, $b \in \mathbb{Z}_{p-1}^*$ arbitrarily and computes

$$\begin{aligned} \alpha &= g^a e^b \bmod p, \\ \beta &= -a\alpha \bmod p - 1, \\ m &= b\alpha \bmod p - 1, \end{aligned}$$

then it can be shown that (α, β) signs m.

For a multisignature, a subset consists of t signers with public keys $e_1, e_2, \ldots, e_t$. To forge a signature for a message m, the actual signer randomly selects $a \in \mathbb{Z}_{p-1}$, $b \in \mathbb{Z}_{p-1}^*$. Let $\alpha = g^a (e_1 e_2 \ldots e_t)^b \bmod p$, $\beta = -a\alpha \bmod p - 1$, and $m = b\alpha \bmod p - 1$. Then, (α, β) is a valid signature of the message m since (α, β) passes the verifications as

$$(e_1 e_2 \cdots e_t)^m = \alpha^\alpha g^\beta \bmod p.$$

The ring-sign and ring-verify of the generalized multisigner ring signature scheme can be defined similar to the generalized ring signature scheme described in Section 3. The detailed description will not be repeated here.

4.1 Security Analysis

The security of the generalized multisigner ring signature depends on the security of 1) the individual ElGamal signature scheme, 2) the ElGamal multisignature scheme,

and 3) the generation of the generalized multisigner ring signature from the ElGamal multisignatures.

The security of 1) the individual ElGamal signature scheme and 2) the ElGamal multisignature have been analyzed in [23] and [24], respectively. For item 3), the only difference between the generalized ring signature scheme and the generalized multisigner ring signature scheme is that for the generalized ring signature, each generalized ring member corresponds to an ElGamal signature, while for the generalized multisigner ring signature, each generalized ring member corresponds to either an ElGamal signature or an ElGamal multisignature. Similar to Theorem 1, we can prove the following theorem.

Theorem 2. *The generalized multisigner ring signature scheme is secure against adaptive chosen-message attack in the random oracle model.*

5 Convertibility of the Proposed Ring Signature

In this section, we will prove that the proposed generalized ring signature scheme is convertible. A ring signature is called *convertible* if it contains the following two algorithms:

- ring-convert. The real signer A_s of a ring signature σ can provide nonrepudiation evidence to a verifier for the originality of the ring signature.
- ring-convert-verify. Given a ring signature (m, σ') and a set of public keys of the ring, the verifier can determine whether (m, σ') is a valid ring signature generated by the ring member A_s.

For a convertible ring signature scheme, besides the two security requirements for ring signature schemes, there is an additional requirement:

- **Unconvertibility against nonsigner.** The advantage that a ring member A_i, $i \neq s$, can successfully convert a ring signature to pass the ring-convert-verify is negligible.

5.1 Convertibility of RSA-Based Ring Signature

The RSA-based ring signature scheme proposed in [1] is not designed to be convertible. In other words, the actual signer is unable to provide nonrepudiation evidence to a verifier for the originality of the ring signature. To achieve convertibility, a modification of the original ring signature scheme was proposed in [16]. However, we point out that the modified scheme cannot ensure nonrepudiation evidence of the actual message signer. Below, we will present a simple improvement so that the RSA-based ring signature can be convertible.

5.1.1 Modification of ring-sign

In order for the actual signer to convert the RSA-based ring signature and provide nonrepudiation evidence for the originality of the ring signature, it was proposed to embed an extra parameter t in [16]. More specifically, it was proposed to substitute $k = h(m)$ with $k = h(m\|t)$ and the ring-sign $(m, \mathcal{S})$ with ring-sign $(m, \mathcal{S}, t)$, where $t = h(x_1, \ldots, x_{s-1}, x_{s+1}, \ldots, x_n, r)$, and $r \in \{0,1\}^b$ is a randomly chosen secret value.

5.1.2 ring-convert

To convert the ring signature, A_s discloses

$$(x_1, \ldots, x_{s-1}, x_{s+1} \ldots, x_n; r).$$

5.1.3 ring-convert-verify

The verification of the convertibility includes two steps:

1. The verifier checks whether $(x_1, \ldots, x_{s-1}, x_{s+1}, \ldots, x_n) \subset \sigma$ is satisfied, where σ is defined in (3). If it is satisfied, then it continues to the next step. Otherwise, the verifier Rejects the signature.
2. Check whether

$$t = h(x_1, \ldots, x_{s-1}, x_{s+1} \ldots, x_n, r).$$

If the verification is successful, then the verifier Accepts it as the real signer. Otherwise, the verifier Rejects.

5.1.4 Problem

From the definition, convertibility requires that the actual ring signer provides nonrepudiation evidence to a verifier for the originality of the ring signature. However, the disclosure of $(x_1, \ldots, x_{s-1}, x_{s+1} \ldots, x_n; r)$ only conveys the information that t can be generated through

$$t = h(x_1, \ldots, x_{s-1}, x_{s+1} \ldots, x_n; r),$$

while the required nonrepudiation evidence cannot be guaranteed. More specifically, for the proposed solution, r cannot provide nonrepudiation evidence of the actual message signer A_s since it cannot be linked to the identity of the actual message signer. In fact, other ring members can also impersonate A_s by concealing r in t and generating the ring signature.

5.1.5 The Proposed Improvement

We can solve this problem by redefining t in the ring-sign algorithm as follows:

$$t = h(\mathrm{sign}_s(m)\|r),$$

where m is the message to be signed, $\mathrm{sign}_s(m)$ is the signature of m generated by A_s, and $r \in \{0,1\}^b$ is a randomly selected secret. The randomness of r can eliminate the previously generated signatures from being reused.

In the ring-convert algorithm, A_s discloses

$$\mathrm{sign}_s(m)\|r.$$

The convertibility can be verified similarly.

Comparing to the original ring signature, only parameter t is new. The security analysis on signer ambiguity and unforgeability of the original scheme still holds true. The unconvertibility against nonsigner is true because nobody is able to generate the signature of message m on behalf of A_s.

5.2 Convertibility of the Generalized Ring Signature Scheme

5.2.1 ring-convert

The generalized (multisigner) ring signature scheme is unique in its convertibility, which enables the actual

message signer to provide nonrepudiation evidence to the originality of the ring signature. The convertibility relies on the knowledge of the discrete logarithm of $l = \log \alpha_s$ that A_s is randomly selected from $\mathbb{Z}_p^*$ to close the ring during the ring signature generation. The computation of l from α_s is a Discrete Logarithm Problem (DLP), which is computationally infeasible.

The actual signer A_s cannot simply disclose l since otherwise the private key of A_s can be computed. The actual message signer A_s can prove that she possess the discrete logarithm l of α_s by signing the message m and generating $\mathrm{sign}_l(m)$, with l as the private key and α_s as the public key. In this way, A_s can provide nonrepudiation evidence to a verifier that she knows the discrete logarithm of α_s.

5.2.2 ring-convert-verify

To verify the convertibility, after successfully verifying the ring signature, the verifier only needs to check that $\mathrm{sign}_l(m)$ is a valid signature for the message m generated with the public key α_s.

6 Conclusion

In this paper, we first introduce a generalized ring signature based on the original ElGamal signature scheme. Comparing to the ring signature construction based on RSA scheme, in the proposed generalized ring signature scheme, all ring members can share the same prime number and all operations can be performed in the same domain. The generalized ring signature scheme is a convertible ring signature that enables the actual message signer to prove to a verifier that only she is capable of generating the ring signature. Moreover, we can construct a generalized multisigner ring signature scheme from a generalized ring signature scheme and multisignatures to increase the level of confidence or to enforce cross-organizational information leaking with high efficiency. The security analysis shows that both schemes are secure against adaptive-chosen message attacks in the random oracle model.

Acknowledgments

The authors would like to thank an anonymous reviewer for the valuable comments on an earlier version of this paper. This research was supported in part by the US National Science Foundation under Award CNS-0716039.

References

[1] R.L. Rivest, A. Shamir, and Y. Tauman, "How to Leak a Secret," *Advances in Cryptology—ASIACRYPT,* 2001.

[2] D. Chaum and E.v. Heyst, "Group Signatures," *Proc. Advances in Cryptology (EuroCrypt '91),* D.W. Davies, ed., vol. 547, pp. 257-265, 1991.

[3] J.L. Camenisch, "Efficient and Generalized Group Signatures," *Proc. Advances in Cryptology (EuroCrypt '97),* W. Fumy, ed., vol. 1233, pp. 465-479, 1997.

[4] J.L. Camenisch and M.A. Stadler, "Efficient Group Signature Schemes for Large Groups," *Proc. Advances in Cryptology (Crypto '97),* B. Kaliski, ed., vol. 1294, pp. 410-424, 1997.

[5] J. Herranz and G. Saez, "Forking Lemmas in the Ring Signatures' Scenario," *Int'l Assoc. Cryptologic Research,* Technical Report 067, http://eprint.iacr.org/2003/067.ps, 2003.

[6] C.P. Schnorr, "Efficient Identification and Signatures for Smart Cards," *Proc. Advances in Cryptology (Crypto '89),* G. Brassard, ed., vol. 435, pp. 239-252, 1989.

[7] J. Xu, Z. Zhang, and D. Feng, "A Ring Signature Scheme Using Bilinear Pairings," *Lecture Notes in Computer Science,* vol. 3325, 2005.

[8] A.K. Awasthi and S. Lal, *A New Proxy Ring Signature Scheme,* http://arxiv.org/PS_cache/cs/pdf/0410/0410010v1.pdf, 2004.

[9] C.-Y. Lin and T.-C. Wu, "An Identity-Based Ring Signature Scheme from Bilinear Pairings," *Proc. 18th Int'l Conf. Advanced Information Networking and Applications (AINA '04),* pp. 182-185, 2004.

[10] W. Cheng, W. Lang, Z. Yang, G. Liu, and Y. Tan, "An Identity-Based Proxy Ring Signature Scheme from Bilinear Pairings," *Proc. Ninth Int'l Symp. Computers and Comm. (ISCC '04),* vol. 1, pp. 424-429, June/July 2004.

[11] C. Gamage, B. Gras, B. Crispo, and A.S. Tanenbaum, "An Identity-Based Ring Signature Scheme with Enhanced Privacy," *Proc. Securecomm and Workshops,* pp. 1-5, Aug./Sept. 2006.

[12] C. Hu and D. Li, "Forward-Secure Traceable Ring Signature," *Proc. Eighth ACIS Int'l Conf. Software Eng., Artificial Intelligence, Networking, and Parallel/Distributed Computing (SNPD '07),* pp. 200-204, July/Aug. 2007.

[13] J. Li, T.H. Yuen, X. Chen, and Y. Wang, "Proxy Ring Signature: Formal Definitions, Efficient Construction and New Variant," *Proc. Int'l Conf. Computational Intelligence and Security (CIS '06),* vol. 2, pp. 1259-1264, Nov. 2006.

[14] C. Hu and D. Li, "A New Type of Proxy Ring Signature Scheme with Revocable Anonymity," *Proc. Eighth ACIS Int'l Conf. Software Eng., Artificial Intelligence, Networking, and Parallel/Distributed Computing (SNPD '07),* vol. 1, pp. 866-868, July/Aug. 2007.

[15] C. Zhang, Y. Liu, and D. He, "A New Verifiable Ring Signature Scheme Based on Nyberg-Rueppel Scheme," *Proc. Eighth Int'l Conf. Signal Processing (ICSP),* 2006.

[16] K.-C. Lee, H.-A. Wen, and T. Hwang, "Convertible Ring Signature," *IEE Proc.-Comm.,* vol. 152, no. 4, pp. 411-414, http://link.aip.org/link/?IPC/152/411/1, Aug. 2005.

[17] S. Goldwasser, S. Micali, and R.L. Rivest, "A Digital Signature Scheme Secure against Adaptive Chosen-Message Attacks," *SIAM J. Computing,* vol. 17, no. 2, pp. 281-308, Apr. 1988.

[18] E. Bresson, J. Stern, and M. Szydlo, "Threshold Ring Signatures and Applications to Ad-Hoc Groups," *Proc. Advances in Cryptology (CRYPTO '02),* vol. 2442, pp. 465-480, 2002.

[19] H. Kuwakado and H. Tanaka, "Threshold Ring Signature Scheme Based on the Curve," *Proc. IEEE Int'l Symp. Information Theory (ISIT '03),* vol. 2442, p. 139, 2003.

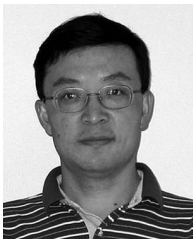
[20] Z.Y. Wu, F. Gwo, J. Tzer, and S. Chen, "A Novel Id-Based Threshold Ring Signature Scheme Competent for Anonymity and Anti-Forgery," *Proc. Int'l Conf. Computational Intelligence and Security (CIS '06),* pp. 1351-1354, Nov. 2006.

[21] Y.-S. Chen, C.-L. Lei, Y.-P. Chiu, and C.-Y. Huang, "Confessible Threshold Ring Signatures," *Proc. Int'l Conf. Systems and Networks Comm. (ICSNC '06),* p. 25, 2006.

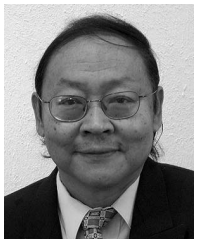
[22] R. Rivest, A. Shamir, and L. Adleman, "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems," *Comm. ACM,* vol. 21, no. 2, pp. 120-126, 1978.

[23] T.A. ElGamal, "A Public-Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms," *IEEE Trans. Information Theory,* vol. 31, no. 4, pp. 469-472, 1985.

[24] L. Harn, "Group-Oriented (t, n) Threshold Digital Signature Scheme and Digital Multisignature," *IEE Proc. Computers and Digital Techniques,* vol. 141, no. 5, pp. 307-313, Sept. 1994.



Jian Ren received the BS and MS degrees in mathematics from Shaanxi Normal University, Xi'an, China, in 1988 and 1991, respectively, and the PhD degree from Xidian University in 1994. He is currently an assistant professor in the Department of Electrical and Computer Engineering, Michigan State University. From 1997 to 1998, he was with Racal Datacom as a security architect. From 1998 to 2002, he was first with Bell-Labs and later with Avaya Labs as a member of technical staff. His research interests include cryptography, network security, sequence design for wireless CDMA communications, E-commerce, and wireless and multimedia communication security. He is a member of the IEEE.



Lein Harn received the BS degree in electrical engineering from the National Taiwan University in 1977, the MS degree in electrical engineering from the State University of New York-Stony Brook in 1980, and the PhD degree in electrical engineering from the University of Minnesota in 1984. In 1984, he joined the Department of Electrical and Computer Engineering, University of Missouri-Columbia as an assistant professor, and in 1986, he moved to Computer Science and Telecommunication Program (CSTP), Department of Computer and Electrical Engineering, University of Missouri, Kansas City (UMKC). While at UMKC, he went on development leave to work in Racal Data Group, Florida for a year. His research interests include cryptography, network security, and wireless communication security. He has published a number of papers on digital signature design and applications and wireless and network security. He has written two books on security. He is currently investigating new ways of using digital signature in various applications.