An Efficient Threshold Anonymous Authentication Scheme for Privacy-Preserving Communications

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

***Abstract*—Anonymous authentication enables any user to be authenticated without being identified. (t, n)-threshold ring signatures, introduced by Bresson *et. al.*, are ring signature schemes that allow a group of t members to jointly sign a message anonymously in a ring of n members. Threshold ring signature schemes provide a nice tradeoff between anonymity and creditability since it allows multiple ring members to sign a message jointly. The complexity in both signature generation and signature verification of the threshold ring signature scheme proposed by Bresson *et. al.* is $\mathcal{O}(n^2)$. They also proposed an efficient threshold ring signature scheme based on an (n, t)-complete fair partition, with complexity $\mathcal{O}(n \log n)$. In this paper, a new efficient (t, n)-threshold ring signature scheme is proposed. This scheme is constructed through a system of t linear equations and n variables, where t is generally a fixed number that is much smaller than n. The proposed threshold ring signature scheme can provide unconditional signer ambiguity, threshold unforgeability and provable security in the random oracle model. The complexity of signature generation and signature verification of the proposed threshold ring signature scheme are $\mathcal{O}(t \log_2^2 t)$ and $\mathcal{O}(n)$, respectively. Furthermore, the length of the threshold ring signature is the same as the regular ring signature introduced by Rivest *et. al.*, which is $2n + 2$, while the length of the threshold ring signature scheme proposed by Bresson *et. al.* is $3n - t + 3$.**

***Index Terms*—Anonymity authentication, threshold ring signature, unconditional secure, unforgeability, random oracle secure, system of linear equations.**

I. Introduction

ANONYMOUS authentication allows a user (or users) to release authenticated information to public or access public content without being identified. Anonymous authentication has found applications in many areas, such as Internet chatting, secret handshake, anonymous information leaking, and so on. While the Internet chatting is very successful, user privacy has become a notable security issue [1]. It is desirable to keep mobile users' identities and whereabouts anonymous [2], [3].

The concept of ring signature was first introduced by Rivest, Shamir and Tauman in 2001 [4] to provide anonymity for the message signer. In a ring signature scheme, the message signers form a ring composed of any possible signers and himself. The actual message signer can then generate a ring signature entirely using only his secret key and the others' public keys without the assistance or even awareness of the other ring members. 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 [5]–[7], but with some fundamental differences. First, unlike group signatures, ring signatures do not require group managers to administrate the joint and revocation of the ring members. The actual message signer has the freedom to select all the ring members and sign whatever messages he like. Second, in a group signature, the group members only look indistinguishable to the verifier but not to the group manager who can revoke the anonymity and recover the real identity of the actual message signer. Recently, a number of signatures schemes have been introduced. A representative example is proxy signatures [8]. In a proxy signature, a delegator gives partial of his signing rights to other parties called proxy signers. However, a proxy signature is fundamentally different from our scheme in that it does not offer signer anonymity.

Ring signatures can be widely applied to many wired/wireless networks to provide anonymous authentication to prevent possible coercing and privacy information leakage, such as e-voting/e-cash, deniable authentication/signature, and so on.

Due to the wide applications of ring signatures, several related ring signature schemes [9]–[11] have been proposed. Particularly interesting is the t-out-of-n, or (t, n)-threshold ring signature scheme proposed by Bresson *et al.* [12], which allows a group of t members to jointly sign a message anonymously in a ring of n ring members without requiring any fixed router or stable links and involving any trusted third party. This is extremely suitable for wireless network scenarios. Threshold ring signature schemes provide a nice tradeoff between anonymity and creditability for the unstable wireless networks since it allows multiple members to sign a message jointly and anonymously. As an example, when a leaked message is signed by multiple *deep throats* jointly, the message will have a much higher creditability than the message that is only signed by a single *deep throat*.

Without threshold ring signature schemes, when multiple signers want to sign the same message, if they use a regular ring signature, multiple ring signatures will be generated. There are two problems in using this approach. First, it is infeasible to determine the exact number of actual message signers. This is because that any signer can generate multiple

Manuscript received November 29, 2012; revised July 16 and October 8, 2012; accepted December 2, 2012. The associate editor coordinating the review of this paper and approving it for publication was G. Xue.

J. Ren is with the Department of Electrical and Computer Engineering, Michigan State University, East Lansing, MI 48864-1226, USA (e-mail: renjian@egr.msu.edu).

L. Harn is with the Department of Computer Science Electrical Engineering, School of Computer Networking, University of Missouri-Kansas City, MO 64110-2499, USA (e-mail: harnl@umkc.edu).

Digital Object Identifier 10.1109/TWC.2012.12.112120

ring signatures and the actual signer of each ring signature is indistinguishable from the other $n-1$ members. Second, the length of the ring signatures and the complexity to verify these ring signatures are in proportional to the number of signers. Threshold ring signatures provide an efficient and coherent way for multiple signers to sign the same message jointly and produce a single threshold ring signature. Any verifier can verify this threshold ring signature at once and determine the exact number of signers. However, the identities of the real signers are totally anonymous to the verifier. Threshold ring signatures differ from threshold signatures [13] in that threshold ring signatures do not need prearrangement for the group formation, while threshold signatures require the group to be prearranged.

In the threshold ring signature scheme proposed in [12], an interpolation polynomial of degree $n-t$ has to be constructed for a threshold ring signature generation and verification. The computational complexity of this process is $\mathcal{O}((n-t)^2) = \mathcal{O}(n^2)$ since n should be much larger than t in order to provide anonymity of the actual message signers. When t is small comparing to n, their construction is very inefficient. Moreover, the threshold ring signature length in this scheme is $3n-t+3 \approx 3n$, longer than the length of the original ring signature $2n+2 \approx 2n$.

To solve this problem, in this paper, a new approach in constructing threshold ring signature scheme is first proposed in literature through a system of t linear equations with n variables, where $t < n$. In the proposed threshold ring signature scheme, to generate a threshold ring signature, one needs to solve a system of t linear equations with t variables. The complexity can be as low as $\mathcal{O}(t \log_2^2 t)$ [14]. The complexity for signature verification is at most $\mathcal{O}(n)$. Therefore, the proposed threshold ring signature is more efficient than the threshold ring signature scheme proposed in [12] in both the threshold ring signature generation and verification. Unlike the threshold ring signature scheme proposed in [12] whose signature size is $3n-t+3 \approx 3n$, in the proposed threshold ring signature, the signature length is the same as the regular ring signature, which is $2n+2 \approx 2n$. Moreover, our novel construction provides unconditional signer-ambiguity, provable security in the random oracle model.

The rest of this paper is organized as follows. In Section II, an overview of the existing ring signatures is given. A new efficient threshold ring signature is introduced in Section III. In Section IV, security analysis of the proposed threshold ring signature is provided. Further analysis of the proposed threshold ring signature is performed in Section V and we conclude in Section VI.

II. Overview of Existing Work

In [4], the concept of the ring signature was first proposed. Suppose that Alice wishes to generate a ring signature of a message m for a ring of n individuals $A_1, A_2, \cdots, A_n$, where the signer Alice is A_s, for some value of $s, 1 \leq s \leq n$. Each ring member A_i has a public key P_i and a corresponding private key S_i.

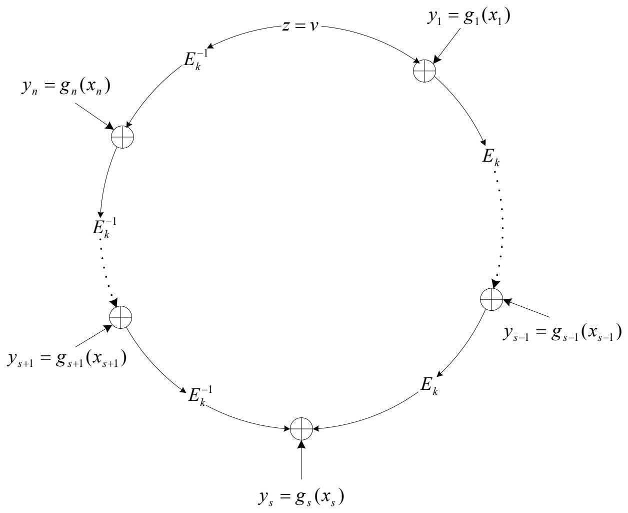


Fig. 1. Ring signatures.

A. Definitions

***Definition** 1 (ring signature):* A *ring signature* scheme consists of the following two algorithms:

- **ring-sign** $(m, P_1, P_2, \cdots, P_n)$: Given a message m and the public keys $P_1, P_2, \cdots, P_n$ of the n ring members, the actual signer A_s can produce a ring signature σ using her own private key S_s.
- **ring-verify** (m, σ): Given a message m and a ring signature σ, which includes the public keys of all possible signers of the ring, a verifier can determine whether (m, σ) is a valid ring signature generated by one of the ring members.

***Definition** 2 (Threshold ring signature):* A threshold ring signature scheme consists of two algorithms:

- **T-ring-sign algorithm**: On an input message m, a ring of n users including n public keys, and the secret keys of t members, it outputs a (t,n)-ring signature σ on the message m. The value of t as well as n public keys of all ring members are included in σ.
- **T-ring-verify algorithm**: On an input message m and a signature σ, it outputs either "True" or "False".

B. Ring Signatures based on RSA

In the ring signature proposed by Rivest, *et. al.*, see Figure 1, each ring member A_i has an RSA public key $P_i = (e_i, n_i)$ which specifies the trapdoor one-way permutation f_i over $\mathbb{Z}_{n_i}$:

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

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 the various ring members have domains of different size, which makes it difficult 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 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, $0 \leq r_i \leq n_i$. Then

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

It is assumed the existence of a publicly defined 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 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.

***Definition 3** (Combining functions):* A combining function $C_{k,v}(y_1, y_2, \cdots, y_n)$ takes as inputs a key k, an initialization value v, and arbitrary values $y_1, y_2, \cdots, y_n \in \{0,1\}^b$. It outputs a value $z \in \{0,1\}^b$, such that for any given $k, v, s, 1 \leq s \leq n$, and fixed values of $y_i, i \neq s$, the function $C_{k,v}$ is a one-to-one mapping from y_s to the output z. This mapping is efficiently solvable. However, it should be infeasible to solve the verification equation without knowledge of the trapdoor information.

In [4], a combining function is proposed as follows:

$$\begin{aligned} z &= C_{k,v}(y_1, \cdots, y_n) \\ &= E_k(y_n \oplus E_k(y_{n-1} \oplus E_k(\cdots \oplus E_k(y_1 \oplus v)))). \end{aligned} \tag{1}$$

For any given s, equation (1) can be rewritten as follows:

$$\begin{aligned} y_s = {} & E_k(y_{s-1} \oplus E_k(\cdots \oplus E_k(y_1 \oplus v))) \oplus \\ & E_k^{-1}(y_{s+1} \oplus E_k^{-1}(\cdots \oplus E_k^{-1}(y_n \oplus E_k^{-1}(z)))). \end{aligned} \tag{2}$$

The ring signature scheme proposed in [4] contains the following two algorithms:

*a) **ring-sign** $(m, P_1, P_2, \cdots, P_n, S_s)$:* Suppose that Alice wishes to sign a message m with a ring signature for the ring of n individuals $A_1, A_2, \cdots, A_n$, where Alice is A_s for some value of $s, 1 \leq s \leq n$. Given the message m to be signed, A_s's secret key $S_s = (d_s, n_s)$, and the sequence of public keys $P_1, P_2, \cdots, P_n$ 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 $1 \leq i \leq n, i \neq s$ 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:

$$C_{k,v}(y_1, y_2, \cdots, y_n) = v.$$

Equivalently, y_s can be solve as follows:

$$\begin{aligned} y_s \quad = \quad & E_k(y_{s-1} \oplus E_k(\cdots \oplus E_k(y_1 \oplus v))) \oplus E_k^{-1}(y_{s+1} \\ & \oplus E_k^{-1}(\cdots \oplus E_k^{-1}(y_n \oplus E_k^{-1}(v)))) \end{aligned}$$

5) **Invert y_s using A_s's trapdoor information:** A_s uses her knowledge of the trapdoor information 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 the $2n+2$-tuple:

$$S = (P_1, P_2, \cdots, P_n; v; x_1, x_2, \cdots, x_n).$$

*b) **ring-verify** $(P_1, P_2, \cdots, P_n; v; x_1, x_2, \cdots, x_n)$:* A verifier can check an alleged signature on the message m as follows:

1) **Apply the trapdoor information:** For $i = 1, 2, \cdots, n$, 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 the fundamental equation

$$C_{k,v}(y_1, y_2, \cdots, y_n) = v.$$

If the ring equation is satisfied, the verifier accepts the signature as valid. Otherwise the verifier rejects.

C. Threshold Ring Signature

Threshold ring signature was first introduced by Bresson *et. al.* in [12]. Assume that t users want to leak some juicy information, so that any verifier will be convinced that at least t users among a selected group of n members vouch for its validity. Simply constructing t ring signatures clearly has no way to guarantee that all the ring signatures have different signers. A (t, n)-threshold ring signature scheme effectively proves that at least t numbers of a group of n members must have actually collaborated to produce the signature, while hiding the membership of a subgroup.

The threshold ring signature proposed in [12] uses Shamir secret sharing scheme [15] to perform a threshold proof. Let m be a message to be signed, the number of possible ring members is n, in which only t members are the actual message signers. The public keys of all ring members are denoted as $P_1, P_2, \cdots, P_n$, while the actual signers are $P_{i_1}, P_{i_2}, \cdots, P_{i_t}$. Define $\mathcal{S} = \{i_1, i_2, \cdots, i_t\}, \overline{\mathcal{S}} = \{i \,|\, 1 \leq i \leq n, , i \notin \mathcal{S}\}$. The threshold ring signature proposed in [12] also contains two algorithms:

*a) **T-ring-sign algorithm**:* On input message m, a ring of n users including n public keys, and the secret key of t members, it outputs a (t, n)-threshold ring signature on the message m as follows:

1) **Compute the symmetric key for E:** $k = h(m)$.
2) **Compute value at origin:** $v = h(P_1, \cdots, P_n)$.
3) **Choose random seeds:** For $i \in \overline{\mathcal{S}}$, select $x_i \in \{0,1\}^b$ randomly, then let $y_i = g_i(x_i)$.
4) **Compute a sharing polynomial:** Compute a polynomial f over $GF(2^b)$ such that $\deg(f) = n-t, f(0) = v$ and For $i \in \overline{\mathcal{S}} : f(i) = E_{k,i}(y_i)$.
5) **Solve the remaining equations:** For $i \in \mathcal{S}$, let $x_i = g_i^{-1}(E_{k,i}^{-1}(f(i)))$.

6) **Output the signature:**

$$(m, P_1, \cdots, P_n, x_1, \cdots, x_n, f).$$

*b) **T-ring-verify algorithm**:* On receiving a tuple

$$(m, P_1, \cdots, P_n, v, x_1, \cdots, x_n, f),$$

the verifying algorithm performs the following steps:

1) **Recover the symmetric key:** $k = h(m)$.
2) **Recover y_i's:** For $i = 1, \cdots, n$, let $y_i = g_i(x_i)$.
3) **Verify the equations:**

$$f(0) \overset{?}{=} h(P_1, \cdots, P_n),$$

$$f(i) \overset{?}{=} E_{k,i}(y_i),\ i = 1, \cdots, n.$$

If all these equations hold true, the verifier accepts the (t, n)-threshold ring signature as valid, otherwise, the verifier rejects it, where $t = n - deg(f)$.

D. Limitation of the Existing Threshold Ring Signature

As can be seen from the above description, for a (t, n)-threshold ring signature proposed in [12], to generate a threshold ring signature, an interpolation polynomial of degree $n - t$ has to be constructed, where n is the total number of ring members, and t is the number of actual message signers. The most efficient construction of such a polynomial is based on the Newton's interpolation polynomial. The computational complexity of this method is $\mathcal{O}((n-t)^2) = \mathcal{O}(n^2)$ since t is generally small and independent of n. To verify a threshold ring signature, the verifier needs to evaluate the interpolation polynomial. The complexity is again $\mathcal{O}(n^2)$. Therefore, both threshold ring signature generation and verification are quite inefficient.

As an example, if we select $n = 50$ and $t = 2$, then an interpolation polynomial of degree $n - t = 50 - 2 = 48$ has to be constructed. This is quite inefficient. To solve this problem, a new efficient threshold ring signature scheme is proposed in the next section. The proposed threshold ring signature scheme is extremely efficient, meanwhile, provably secure in the random oracle model.

To reduce the complexity, Bresson *et. al.* proposed an efficient threshold ring signature scheme based on an (n, t)-complete fair partition. However, the complexity is still $\mathcal{O}(n \log n)$.

III. A Threshold Ring Signature Scheme with Linear Efficiency

In this section, a new efficient threshold ring signature scheme is introduced. Our description of the new threshold ring signature scheme is based on the RSA signature scheme described in Section II-B.

For a regular ring signature scheme with n possible signers, the actual message signer forgers a signature for each of the $n - 1$ arbitrarily selected ring members. In order to glue the signature ring, the actual message signer has to sign the message that is required to glue the ring signature using his trapdoor information. This process ensures the involvement of a real signer in generating the ring signature.

For a (t, n)-threshold ring signature scheme, to create a message signature, we need to ensure that at least t trapdoors are utilized. In other word, the actual message signers are at least t. Simply constructing t regular ring signatures clearly has no way to guarantee that all the ring signatures are signed by different users. Therefore, some kind of cohesive relationship is need to ensure that the number of actual message signers is at least t.

In this section, the aforementioned problem will be tackled through a system of t linear equations with n variables, where $t < n$. In fact, it is well-known that for a system of linear equations, if the number of independent equations t is smaller than the number of variables n, then for an arbitrary assignment of any $n - t$ selected variables, the rest t variables can be uniquely solved. This means that any arbitrarily selected t variables can be represented by the rest $n - t$ variables. Since $n - t$ variables can be assigned values randomly, regardless of which $n - t$ variables are selected and what value they are assigned, the system of linear equations has exactly $(2^b)^{n-t}$ different solutions, and all of them can be chosen with equal probability without depending on any complexity-theoretic assumptions or on the randomness of the oracle.

***Definition** 4 (Threshold ring signature combining functions):* The proposed (t, n)-threshold combining functions $C_{k,j}(y_1, \cdots, y_n), j = 1, \cdots, t$, take as input n arbitrary values $y_1, \cdots, y_n \in \mathbb{Z}^*_{p-1}$, where p is a large prime number. The output values are $h(m, j)$, defined as

$$\begin{aligned} h(m, j) &= C_{k,j}(y_1, \cdots, y_n) \\ &= \sum_{i=1}^{n} i^j \cdot E_k(y_i) \bmod p,\ j = 0, 1, \cdots, t-1, \end{aligned} \quad (3)$$

For each $j \in \{0, 1, \cdots, t-1\}$, the function $C_{k,j}$ is a one-to-one mapping from $(y_1, \cdots, y_n)$ to $h(m, j)$ that is efficiently solvable. However, it should be infeasible for an adversary to solve $x_1, \cdots, x_n$ from

$$C_{k,j}(g(x_1), \cdots, g(x_n)) = h(m, j),$$

without knowledge of the trapdoors.

For equation (3) that contains t equations with n variables $(y_1, y_2, \cdots, y_n)$, or $(E_k(y_1), E_k(y_2), \cdots, E_k(y_n))$, where $t < n$, if we randomly select the values of any $n - t$ y_i's, then the rest t y_i's can be uniquely computed from the system of linear equations (3) based on the arbitrarily selected $n - t$ y_i's. Without loss of generality, suppose $i_1, i_2, \cdots, i_n$ is a permutation of $1, 2, \cdots, n$ and $y_{i_{t+1}}, y_{i_{t+2}}, \cdots, y_{i_n}$ are the randomly selected values. Then equation (3) can be rewritten as equation (4).

The coefficient matrix V of equation (4) given below

$$V = \begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ i_1 & i_2 & i_3 & \dots & i_t \\ i_1^2 & i_2^2 & i_3^2 & \dots & i_t^2 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ i_1^{t-1} & i_2^{t-1} & i_3^{t-1} & \dots & i_t^{t-1} \end{bmatrix},$$

is a Vandermonde matrix of dimension $t \times t$. The determinant

$$\begin{cases} E_k(y_{i_1}) + E_k(y_{i_2}) + \cdots + E_k(y_{i_t}) = h(m,0) - \sum_{l=t+1}^{n} E_k(y_{i_l}) \bmod p \\ i_1 E_k(y_{i_1}) + i_2 E_k(y_{i_2}) + \cdots + i_t E_k(y_{i_t}) = h(m,1) - \sum_{l=t+1}^{n} l \cdot E_k(y_{i_l}) \bmod p \\ \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\ i_1^{t-1} E_k(y_{i_1}) + i_2^{t-1} E_k(y_{i_2}) + \cdots + i_t^{t-1} E_k(y_{i_t}) = h(m,t-1) - \sum_{l=t+1}^{n} l^{t-1} \cdot E_k(y_{i_l}) \bmod p \end{cases} . \quad (4)$$

of matrix V can be computed as:

$$\det(V) = \prod_{1 \le v < u \le t} (i_u - i_v).$$

Since $i_1, i_2, \cdots, i_t$ are all different integers from $\{1, 2, \cdots, n\}$, we have $\det(V) \neq 0$. In other words, the coefficient matrix has full rank t. Therefore, $y_{i_1}, y_{i_2}, \cdots, y_{i_t}$ are uniquely solvable.

This concludes that for a system of linear equations (4), any t arbitrarily selected y's can be uniquely expressed by the rest $n-t$ y's, therefore, they could possibly be generated using the knowledge of trapdoors. Moreover, since these $n-t$ y's can assume any values with equal probability, the system of linear equations (4) has exactly $(2^b)^{n-t}$ different solutions.

Now let us describe the proposed (t,n)-threshold ring signature scheme below.

*a) **T-ring-sign algorithm**:* On an input message m, a ring of n users including n public keys, and the participant of t members, it outputs a (t,n)-threshold ring signature on the message m as follow:

1) **Compute the symmetric key for E:** $k = h(m)$.
2) **Pick random x_i's for $i \in \overline{\mathcal{S}}$:** The purpose of this step is to randomly forge a signature for some input x_i for each of the $n-t$ non-signer ring members and then computes
$$y_i = g_i(x_i).$$
3) **Compute $h(m,j)$:** Compute $h(m,j)$, for $j = 1, 2, \cdots, t$.
4) **Solve the y_i over $GF(2^b)$ for $i = 1, \cdots, t$:** The actual message signers solve each y_i from the proposed threshold combining functions:
$$C_{k,j}(y_1, \cdots, y_n) = h(m,j),\ j = 1, 2, ..., t$$
defined in equation (4).
5) **Invert the actual signers' trapdoor permutations:** Each actual signer uses the knowledge of his/her trapdoor in order to invert g_i on $y_i, i \in \mathcal{S}$, to obtain x_i:
$$x_i = g_i^{-1}(y_i).$$
6) **Output the signature:**
$$(m, t, P_1, \cdots, P_n, x_1, \cdots, x_n).$$

*b) **T-ring-verify algorithm**:* On receiving a tuple
$$(m, t, P_1, \cdots, P_n, x_1, \cdots, x_n),$$
a verifier can verify an alleged signature on the message m as follows:

1) **Recover the symmetric key for E:** $k = h(m)$.
2) **Apply the trapdoor permutations:** For $i = 1, 2, \cdots, n$, the verifier computes
$$y_i = g_i(x_i).$$
3) **Verify the equations:** The verifier checks that the x_i satisfy the fundamental equation (3). If all the t equations are satisfied, the verifier accepts the signature as a valid (t,n)-threshold ring signature. Otherwise the verifier rejects.

IV. Security Analysis

In this section, signer ambiguity and threshold unforgeability of the proposed threshold ring signature will be discussed.

***Theorem 1:** (Unconditional signer ambiguity):* The proposed threshold ring signature can achieve unconditional signer ambiguity.

Proof: Let t be the actual number of signers. It is straightforward that any t distinct ring members are able to produce a valid threshold ring signature. The identity of the t message signers are unconditionally protected with the proposed threshold scheme. This is because that regardless of the signers' identity, the signing algorithm chooses a system of t linear equations with n variables. Therefore, $n-t$ variables can be chosen at random, while the rest t variables can be uniquely computed from the rest $n-t$ variables. This holds independently from the signing group's membership and all options can be chosen with equal probability without depending on any complexity-theoretic assumptions. ■

***Theorem 2:** (Threshold unforgeability):* The proposed threshold ring signature scheme is existentially unforgeable against adaptive chosen message attacks in the random oracle model.

Proof: We have to prove that a correct signature is necessarily produced by at least the claimed number of users, say t. To prove this theorem, we only need to prove that in the random oracle model, any forgery algorithm $\mathcal{A}$ with non-negligible probability for a new threshold ring signature for m' by analyzing polynomially many threshold 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 non-negligible probability.

The main idea is that with non-negligible probability, a forger outputs a forgery for which the system of linear equations is such that at least t values x_i were asked to the

E^{-1} oracle. If the forger has corrupted at most $t-1$ users, then with high probability, we can use it to invert one of the corresponding one-way permutations.

Assume that there exists a forging algorithm $\mathcal{A}$, that succeeds in creating a forgery with non-negligible probability. More specifically, algorithm $\mathcal{A}$ gets a subset $I_0 \subset [1, n]$ is of cardinality $t-1$ composed of the corrupted players. Without loss of generality, we suppose that the first $t-1$ users are the corrupted users. The matching pairs of their private/public keys are $P_1/S_1, \cdots, P_{t-1}/S_{t-1}$. The rest $n-t$ users' public key are $P_t, \cdots, P_n$, where each P_i specifies a trapdoor one-way function. Their private keys will not be used. $\mathcal{A}$ is also given oracle access to h, E, E^{-1} and to a ring signing oracle. It can work adaptively, querying the oracles at arguments that may depend on previous answers. Eventually, it produces a valid ring signature on a new message that was not presented to the signing oracle, with a non-negligible probability. We show that $\mathcal{A}$ can be turned into an algorithm $\mathcal{B}$ which inverts one of the trapdoor one-way functions g_i on random inputs y with non-negligible probability.

Algorithm $\mathcal{B}$, on input $g_t, \cdots, g_n$ and a random value $y \in \{0,1\}^b$, uses algorithm $\mathcal{A}$ on input $(g_t, \cdots, g_n)$ as a black-box while simulating its oracles, in order to find a value $g_i^{-1}(y_0)$ for some $i \in \{t, \cdots, n\}$.

In order to make the probability of satisfying the ring equation becomes non-negligible, algorithm $\mathcal{B}$ simulates the random oracle h with the message that it is actually going to forger in a straightforward way, answering a random value for each new query, and maintaining a list of already queried messages. It also simulates the symmetric encryptions E_k, in such way that it is consistent with a random permutation. For simplicity, and without loss of generality, we do not allow $\mathcal{A}$ to make a query $E_k(x)$ if it has already got x as answer for a query $E_k^{-1}(y)$.

Assume that, with non-negligible probability, algorithm $\mathcal{A}$ forges the j'th message that it sends to the oracle h. We denote this message by m'. Algorithm $\mathcal{B}$ begins by guessing randomly this index j. Note that algorithm $\mathcal{B}$ guesses the correct value with non-negligible probability since algorithm $\mathcal{A}$ makes in total at most polynomially many queries to the oracle h.

Algorithm $\mathcal{B}$ simulates the signing oracle by providing a random vector $(h(m,1), \cdots, h(m,t),\ x_1, \cdots, x_n)$ as a threshold ring signature to any query m. It then adjusts the random answers to queries of the form $E_{h(m)}$ and $E_{h(m)}^{-1}$, to support the correctness of the ring equation (3) for these messages. Note that $\mathcal{A}$ cannot ask oracle queries that will limit $\mathcal{B}$'s freedom of choice, before providing m to the signing oracle. Algorithm $\mathcal{B}$ must remember whether x was answered on a E^{-1} query for y, or the opposite. Algorithm $\mathcal{B}$ answers the encryption and decryption-queries using a random value for each new query. However, on the new decryption-query for x_0, algorithm $\mathcal{B}$ answers with y_0, which is uniformly distributed over $\{0,1\}^l$.

It remains to simulate a signing oracle for an arbitrary subgroup of signers, simply by choosing randomly the components of the signature, and by adapting its answers to the E or E^{-1}-queries, for $k = h(m')$. Recall that the goal of algorithm $\mathcal{B}$ is to compute $x_i = g_i^{-1}(y)$, for some i between t and n. The basic idea is to slip this value y as the "gap" between the output and the input values along the ring equation of the final forgery, which forces $\mathcal{A}$ to provide the x_i in the generated signature. This basic idea is carried out in the following way.

We note that with overwhelming probability E_k and E_k^{-1} are not constrained up to the point where $\mathcal{A}$ queries the oracle h with query m'. Thus, $\mathcal{B}$ will do the following immediately after $\mathcal{A}$ queries the oracle h with query m'. Notice that $\mathcal{A}$ must query the oracles E_k or E_k^{-1} about each one of the n symmetric encryptions along the forgery threshold ring signature. Since all queries for the uncorrupted users, except for one $i \in \{t, \cdots, n\}$, can be answered randomly, unless the values of this query has already be determined by $\mathcal{B}$, in which case it is answered with the predetermined value. Therefore, algorithm $\mathcal{B}$ can simulate answers to these queries that would yield the value $g_i^{-1}(y)$ for some $i \in \{t, \cdots, n\}$.

Finally, $\mathcal{B}$ can simulate a signing oracle for an arbitrary subgroup of signers, simply by choosing randomly the components of the signature, and by adapting its answers to the E or E^{-1}-queries.

$\mathcal{A}$ is allowed to query for the secret key of a player, at any time, obtaining up to $t-1$ secret keys. Such queries are answered in a straightforward way, except if asked to a player $P_j, j \notin I_0$. In the latter case, since y is a random value, the simulated oracles E_k and E_k^{-1} cannot be distinguished from the real oracles, and therefore, with non-negligible probability, adversary $\mathcal{A}$ will output a t-forgery threshold ring signature $(m', t, P_1, \cdots, P_n, x_1, \cdots, x_n, r)$ to a message m'. Moreover, with non-negligible probability there exists $i \in \{t, \cdots, n\}$ such that $g_i(x_i) = y$, as desired. ■

V. Further Analysis of the Proposed Threshold Ring Signature Scheme

In this section, the proposed (t, n)-threshold ring signature will be analyzed and compared with the threshold ring signature proposed in [12]. We found that the proposed threshold ring signature has two major advantages. A table that compares the complexity and performance of the two schemes is provided in Table 1.

1) **Complexity**: For the existing threshold ring signature scheme, we already discussed in Section II-D that the complexity for ring signature generation is $\mathcal{O}(n^2)$. While for the proposed threshold ring signature scheme, to generate a threshold ring signature, one needs to solve a system of t linear equations with t variables. The complexity is $\mathcal{O}(t \log_2^2 t)$ [14]. Keep in mind that in order to provide anonymity of the actual message signers, the number of n should be much larger than the number of t, therefore, $n-t$ should be much larger than t. Hence, the proposed threshold ring signature is more efficient in signature generation.

   It is also pointed out in Section II-D that the complexity for the verification of the existing threshold ring signature scheme is $\mathcal{O}(n^2)$ for the original scheme, or $\mathcal{O}(n \log n)$ for the scheme based on an (n, t)-complete fair partition. While in the proposed threshold ring signature scheme, to verify a threshold ring signature, the verifier only needs to check whether equation (3) holds. The complexity of this procedure is at most $\mathcal{O}(n)$.

TABLE I
COMPLEXITY AND EFFICIENCY COMPARISON, WHERE $t \ll n$, AND IS GENERALLY A FIXED NUMBER, WHILE n IS A LARGE VARIABLE.

	Complexity in signature generation	Complexity in signature verification	Size of signature	An example: $n = 50, t = 2$
Bresson's original	$\mathcal{O}(n^2)$	$\mathcal{O}(n^2)$	$3n$	Need to compute a polynomial of degree 48
Bresson's modified	$\mathcal{O}(n \log n)$	$\mathcal{O}(n \log n)$	$3n$	Need to compute a polynomial of degree 48
Proposed	$\mathcal{O}(t \log_2^2 t)$	$\mathcal{O}(n)$	$2n$	Need to solve a system of 2 equations with 2 variables

Therefore, the proposed threshold ring signature is much more efficient in both threshold ring signature generation and signature verification than the existing threshold ring signature scheme proposed in [12].

2) **Threshold ring signature length**: For the threshold ring signature proposed in [12], the length of the threshold ring signature is $3n - t + 3 \approx 3n$ tuples, while for the proposed threshold ring signature scheme, the length of the threshold ring signature is the same as the regular ring signature introduced in [4], which is $2n + 2 \approx 2n$ tuples. The difference is about n tuples, or nb bits since each tuple can be up to b bits in size.
3) **Feasibility**: In our proposed scheme, we do not specify the public-key algorithms, which means that our proposed scheme works for all public-key systems. The efficiency and feasibility analysis of the existing research on public-key algorithms [16]–[19] demonstrates that our proposed solution can be implement in real-world devices, including for example, smart cards, mobile devices, etc.

VI. CONCLUSION

In this paper, a new efficient (t, n)-threshold ring signature scheme is proposed based on a system of t linear equations with n variables, where t is the number of actual message signers and n is the total number of ring members. The proposed threshold ring signature can achieve unconditional signer ambiguity and is existentially unforgeable against adaptive chosen message attacks in the random oracle model. The proposed threshold ring signature scheme has two major advantages over the existing threshold ring signature scheme. First, the proposed threshold ring signature has a much lower computational complexity in both threshold ring signature generation and signature verification. Secondly, the threshold ring signature length of the proposed ring signature scheme is the same as the regular ring signature length, which is $2n+2$.

ACKNOWLEDGMENT

This research was supported in part by NSF grants CNS-0845812, CNS-1117831, CNS-1217206, ECCS-1232109.

REFERENCES

[1] Online Victimization, "A report on the nation's youth," by the Crimes Against Children Research Center (founded by the U.S. Congress). Available: http://www.missingkids.com/en_US/publications/NC62.pdf.

[2] G. Yang, Q. Huang, D. S. Wong, and X. Deng, "Universal authentication protocols for anonymous wireless communications," *IEEE Trans. Wireless Commun.*, vol. 9, no. 1, pp. 168–184, Jan. 2010.

[3] L. Harn and J. Ren, "Generalized digital certificate for user authentication and key establishment for secure communications," *IEEE Trans. Wireless Commun.*, vol. 10, no. 7, July 2011.

[4] R. Rivest, A. Shamir, and Y. Tauman, "How to leak a secret," in *Advances in Cryptology–ASIACRYPT*, Lecture Notes in Computer Science, vol. 2248/2001. Springer, 2001.

[5] D. Chaum and E. van Heyst, "Group signatures," in *Advances in Cryptology - EUROCRYPT*, Lecture Notes in Computer Science, vol. 547, pp. 257–265, 1991.

[6] J. L. Camenisch, "Efficient and generalized group signatures," in *Advances in Cryptology - EUROCRYPT*, Lecture Notes in Computer Science, vol. 1233, pp. 465–479, 1997.

[7] J. L. Camenisch and M. A. Stadler, "Efficient group signature schemes for large groups," in *Advances in Cryptology - Crypto'97*, Lecture Notes in Computer Science, vol. 1294, pp. 410–424, 1997.

[8] M. Mambo, K. Usuda, and E. Okamoto, "Proxy signatures," in *Proc. 1995 Symposium on Cryptography and Information Security*, pp. 11–17.

[9] J. Herranz and G. Saez, "Forking lemmas in the ring signatures' scenario," Technical Report 067, International Association for Cryptologic Research. Available: http://eprint.iacr.org/2003/067.ps, 2003.

[10] C. P. Schnorr, "Efficient identification and signatures for smart cards," in *Advances in Cryptology - Crypto'89*, Lecture Notes in Computer Science, vol. 435, pp. 239–252, 1989.

[11] M. Abe, M. Ohkubo, and K. Suzuki, "1-out-of-n signatures from a variety of keys," in *ASIACRYPT*, Lecture Notes in Computer Science, vol. 2501, pp. 415–432, 2002.

[12] E. Bresson, J. Stern, and M. Szydlo, "Threshold ring signatures and applications to ad-hoc groups," in *Proc. 2002 Advances in Cryptology*, Lecture Notes in Computer Science, vol. 2442, pp. 465–480, 2002.

[13] L. Harn, "Group-oriented (t, n) threshold signature and multisignature," *IEE Proc.-Computers and Digital Techniques*, vol. 141, no. 5, pp. 307–313, Sep. 1994.

[14] L. Li, "On the arithmetic operational complexity for solving Vandermonde linear equations," *Japan J. Industrial and Applied Mathematics*, vol. 17, no. 1, pp. 15–18, Feb. 2000.

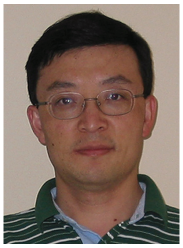
[15] A. Shamir, "How to share a secret," *Commun. of the ACM*, vol. 22, no. 11, pp. 612–613, 1979.

[16] H. Wang, S. Sheng, C. Tan, and Q. Li, "Comparing symmetric-key and public-key based security schemes in sensor networks: a case study of user access control," in *Proc. 2008 IEEE ICDCS*, pp. 11–18.

[17] M. J. Wiener, "Performance comparison of public-key cryptosystems," vol. 4, no. 1, pp. 1–5, 1998.

[18] H. Handschuh and P. Paillier, "Smart card crypto-coprocessors for public-key cryptography," vol. 4, no. 1, pp. 6–10, 1998.

[19] P. Karu and J. Loikkanen, "Practical comparison of fast public-key cryptosystems. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.96.5694.



Jian Ren received the BS and MS degrees both in mathematics from Shaanxi Normal University, and received the Ph.D. degree in EE from Xidian University, China. He is an Associate Professor in the Department of ECE at Michigan State University. Dr. Ren was the Leading Secure Architect at Avaya Lab, Bell Lab and Racal Datacom in security architecture and solution development. His current research interests include cryptography, network security, energy efficient sensor network security protocol design, privacy-preserving communications, and cognitive networks. He is a recipient of the US National Science Foundation Faculty Early Career Development (CAREER) award in 2009. He is a senior member of the IEEE.

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