Confessible Threshold Ring Signatures[1]

Yu-Shian Chen*, Chin-Laung Lei†, Yun-Peng Chiu‡ and Chun-Ying Huang§
Department of Electrical Engineering
National Taiwan University
Taipei, Taiwan
Email: {*ethan, ‡frank, §huangant}@fractal.ee.ntu.edu.tw †Email: lei@cc.ee.ntu.edu.tw

Abstract— **We present two threshold ring signature schemes with different properties. One focuses on the confessibility (or signer verifiability) and the denouncibility properties. The other focuses on the threshold-confessibility. Our schemes are built on generic ring signature schemes and can be easily adapted to most existing ring signature schemes. Based on the former works, we also construct a realization of our schemes as an example. We prove that our schemes are secure in the random oracle model.**

I. Introduction

Suppose that Alice, Bob, and Charles are employees of a bank and they plan to leak a juicy fact to the president Philip about the embezzlement of their manager. To hide their identities, they generate a $(3, 8)$ threshold ring signature. After verifying the fact of the embezzlement, Philip would like to offer a premium to those reporters. Here comes the problems. In scenario I, Alice want to confess and claim the premium while the other two tended to be kept hidden. How do Alice confess to Philip that she was one of the three actual signers? Moreover, is Alice able or not to denounce Bob and Charles to Philip? In scenario II, suppose that Alice, Bob, and Charles have agreed that the confession to Philip must be done by at least two of them. How to achieve this goal using the *double* threshold ring signature?

We will discuss the *verifiability* of the actual signer identities in threshold ring signatures in both the scenarios. We define three new security notions to differentiate from the ambiguity in verifiability [3], [4], [12], [18]. Notably, in our scenario II, we offer a ring signature scheme with "double" threshold structures.

A. Related Work

Ring Signatures. Rivest *et al.* [15] introduced the notion of ring signatures. A ring signature scheme is a group signature scheme without manager and prearrangement. To produce a ring signature, the *actual signer* declares an arbitrary set of *innocent signers* to form a group of *posssible signers* including itself. In particular, the actual signer is able to compute the signature entirely by itself while the innocent signers may be completely unaware. Any recipient can only verify that someone in the group had generated the signature but not ascertain who it is.

Threshold Ring Signatures. A (t, n) threshold ring signature scheme is a ring signature scheme where each signature is a proof that at least t of the n possible signers are the actual signers. Threshold or general access structure of ring signatures have been discussed in [2], [5], [9].

Separable Ring Signatures. A ring signature scheme is said to be separable if all participants can choose their keys independently with different parameter domains and for different types of signatures.

Linkable Ring Signatures. Liu *et al.* [10] introduced the notion of linkability, which means that anyone can determine whether two ring signatures on the same message are signed by the same group member or not.

Verifiable Ring Signatures. Lv and Wang [13] formalized the notion of verifiable ring signatures, which allow the actual signer to prove to the recipient that he had generated the signature if he wishes. The recipient can also verify if the claim from the signer is true or not. In [12], Lv *et al.* applied their former work to the ring authenticated encryption scheme with verifiability property. Later in [3], Cao *et al.* found some weakness in Lv *et al.*'s scheme which cannot achieve signer-verifiability and recipient-verifiability properties. In [4], they proposed an Identity-based (ID-based) ring authenticated encryption whose signer-verifiability is obtained by publishing the random seed. Another ring signature scheme based on discrete logarithm cryptosystems with the *signer-admission* property which is equivalent to the *verifiability* property appeared in [18].

Actually the original ring signature scheme is implicit verifiable. In [15], the authors mentioned that the actual signer can prove to any recipient that a signer is innocent by publishing the random seed used to generated the innocent signer's part of the signature. To prove its own involvement, the actual signer has to publish the seed used to generate all innocent signers' parts of the signature. Nevertheless, the drawback is that to confess its involvement, the actual signer must prove all other signers' are innocent.

The *linkability* property in [11], [17] also implies verifiability. If the actual signer signed on the same message twice, its identity is likely to be exposed. Thus the system administrator can ask all possible signers to simulate the signature generation and then identify all the actual signers. The drawback is that the actual signers will loss their spontaneity of confession.

The term *verifiability* is multivalent in [3], [4], [12], [18]. There are at least the three following meanings. (1) *signature*

[1] This work is supported in part by the National Science Council under the Grants NSC 95-3114-P-001-001-Y02 and NSC 95-2218-E-002-005, and by the Taiwan Information Security Center (TWISC), National Science Council under the Grants No. NSC 94-3114-P-001-001Y and NSC 94-3114-P-011-001.

0-7695-2699-3/06/$20.00 (c) IEEE

verifiability: Given the public keys of the ring members, any recipient can determine if the signature is signed by some members; (2) *signer verifiability or signer-admission*: If the actual signer is willing to confess to a recipient that he really generated the signature, the recipient can correctly determine whether it is true; (3) *recipient verifiability*: Anyone can be convinced who is the designated recipient by the actual signer or the legal recipient.

The verifiability properties in above works are either restricted to non-threshold ring signatures or based on some specific cryptosystems such as RSA, DL, or ID-based.

B. Our Contributions

In this paper, we will discuss the property of verifiability and then extend it to a threshold fashion. First, to discriminate the differences among the ambiguity of verifiability [3], [4], [12], we introduce three new security notions.

Confessibility: The actual signer is able to prove to any recipient that he is one of the signers who has actually signed the signature without disclosing his private key. This property is equivalent to *signer verifiability*.

Denouncibility: An actual signer U_A can give possible signer U_B (or anyone even not in the group) the authority to denounce U_A. Then, U_B can prove to any recipient U_A's involvement without disclosing both of their private keys.

Threshold-Confessibility: In a (t, n) threshold ring signature scheme, t' $(t' \leq t)^2$ actual signers or more are able to jointly prove to any recipient that they had involved in generating the signature without disclosing their private keys nor denouncing other signers. The actual signers whose number is less than the threshold t' will neither convince any recipient anyone's involvement nor denounce any possible signers.

Basically, due to the anonymity of ring signatures, the actual signers have to hold some secret information, which we call *"voucher"* to prove that they are not innocent signers. While confessing or denouncing, the prover is aiming to uncover the secret *voucher* of the target actual signer who may be itself or others.

We proposed two threshold ring signature schemes which satisfies the requirements. In scenario I which requires *confessibility* and *denouncibility*, the actual signers are able to independently confess to any recipient their involvement in generating signature. Moreover, with exchange of their secret *vouchers*, the actual signers are able to denounce each other's involvement. In scenario II which requires *threshold-confessibility*, by combining the technique of distributed key generation, the actual signers are able to jointly confess their involvements in a threshold fashion.

We do not discuss *threshold-denoucibility* because it seems a paradox that some actual signer gives the other signer the authority (signed *voucher*) to denounce itself whereas they had agreed on a threshold scheme for confession.

The term *threshold* in our schemes has meanings in two aspects. One means that the number of the actual signers is threshold in generating signature, and the other indicates that the number of the actual signers is threshold in confessing.

Our schemes provide extensible functions for existing threshold ring signature schemes. Except the subroutine DKG used to manipulate *voucher* in scenario II, our schemes are not based on specific cryptosystems for the core ring signature schemes. We also modify the schemes in [8], [10] to illustrate our extension. Following the same heuristic, most present threshold ring signature schemes can be easily integrated with our schemes.

Organization: The remainder of this paper is organized as follows. In Sec. II, we define the notations and settings of our schemes and introduce a distributed key generation protocol as building block for scenario II. In Sec III, we present our generic schemes and later provide an realization in Sec IV. The security analysis is presented in Sec V. We conclude in Sec. VI.

II. Preliminaries

A. Notations and Setup

Suppose that there is a (t, n) threshold ring signature. Without loss of generality, we assume that $\{U_m | m = 1, 2, ...n\}$, $\{U_i | i = 1, 2, ...t\}$, and $\{U_j | j = t + 1, ...n\}$ denote the set of possible signers, actual signers, and innocent signers, respectively. Let $Sig_m(.)$ denote the individual signature algorithm of signer U_m.

The "core" threshold ring signature schemes is not to base on specified cryptosystems since possible signers may use different types of keys. As the idea in [16], we denote $\mathcal{G}_m$ as the trapdoor one-way permutation of the possible signer U_m. $\mathcal{G}_m$ maybe a encryption algorithm, signature algorithm, or other operations. The reversal algorithm $\mathcal{G}_m^{-1}$ should be computed only by U_m.

Five public known one-way hash functions are used in our schemes: H_0, H_1, H_2, H_3 and H_4. We do not define the practical mapping of these hash functions since their configuration depend on the real ring signature scheme implemented.

B. Distributed Key Generation Protocol

Distributed key generation allows a set of n members to jointly generate a pair of public and private keys. The public key is open and the private key is maintained as a (virtual) shared secret in (t', t) threshold scheme where $t' \leq t$. We simplify the protocol in [6] to the DKG protocol, which is a building block for our scheme. The DKG protocol works as follows: t player jointly generate a share secret key d. Each player U_i only knows his shadow β_i and the public key v_s $(v_s = g^d)$ but no one knows d. A number of t' or more players can publish their shadows and jointly reconstruct d. For lack of space, we make our DKG scheme as succinct as possible. For more discussions and variants about distributed key generation, see [6], [14].

The DKG protocol is used to handle the *vouchers* in scenario II. This subroutine is based on discrete logarithm.

[2] Note that t' denotes the number of actual signers who are willing to confess. Thus it is no more than the original number of actual signers t . For simplicity, we do not consider the situation that t' is higher than t, which means that there are more than t actual signers intentionally to generate only a t-threshold ring signature, in such case t' will be possibly more than t.

Let p and q be large primes such that $q|p-1$ and $q \geq 2^l$, where l is the security parameter of the scheme. Let g be an element of Z_p^* with order q. To be concise, we omit modular operation notation in most place.

DKG protocol

Suppose that members in $\{U_i\}$ agree on generating a shared key in (t', t) threshold. They run the following protocol.

1: Each U_i randomly chooses $\beta_i \in Z_q$ and keeps it secret. Now they have virtually formed a (t', t) secret sharing scheme whose t'-degree polynomial f' satisfying $f'(i) = \beta_i$ but no one knows f'.

2: Each U_i computes $u_i = g^{\beta_i}$ and broadcasts u_i.

3: Each U_i computes

$$v_s = \prod_{0<i\leq t} u_i^{\lambda_i(0)} \tag{1}$$

where $\lambda_i(z)$ is the Lagrange coefficient:

$$\lambda_i(z) = \prod_{j\neq i}(z-j)/(i-j) \tag{2}$$

such that $v_s = g^d = g^{f'(0)}$.

The $v_s = g^d$ is the public key while the shared secret key d is unknown to any one under the assumption of discrete logarithm problem. Following the same heuristic, this protocol can be extended to elliptic curve or other possible cryptosystem with homomorphism property, i.e $E_k(a) \oplus E_k(b) = E_k(a+b)$. For the simplicity of explanation, in this paper, we only focus on discrete logarithm.

III. The Generic Confessible Threshold Ring Signature Schemes

As described in Section I, scenario I possesses the property of *confessibility* and *denouncibility* and scenario II possesses the property of *threshold-confessibility*.

A. Basic Idea

In a threshold ring signature scheme, every possible signer U_m corresponds to a point $f(m)$ of the polynomial f. The basic approach of our schemes is to extend the degree of this polynomial f with extra point $f(r_k)$ $(0 < k \leq t)$. Each extra point is corresponding to one actual signer. Each actual signer decides its pair value $f(r_k)$ which is derived from its own *voucher*. But unlike regular threshold ring signature, the domain value r_k should not be related to the identity of any signers explicitly, otherwise the anonymity will be lost. So the key principle is that the definition and choice of r_k is publicly well-known such that no one can determine the choice of r_k. A simple example is $r_k = H(k, \{U_m\})$.

Since r_k is predefined, the actual signers have to agree on how to assign r_k to each one. A trivial way is by identity order, but random order is reasonably more secure. We denote (i) as a re-ordered index of identity i. For example, a five-member group indexed as $[1, 2, 3, 4, 5]$ is re-ordered to $[2, 5, 1, 4, 3]$, then the new index of the second signer (2) is mapped to 5 and (3) is mapped to 1.

B. Scenario I

Given a message M, the actual signers $\{U_i\}$ run the following algorithm to generate a confessible (t, n) threshold ring signature.

Signature Generation

1: Without loss of generality, an arbitrary actual signer (for example, U_1) prepares the signature on behalf of the other actual signers by performing: (a) randomly chooses a_j for U_j and computes $c_j = \mathcal{G}_j(a_j)$, and (b) broadcasts a_j and c_j to $\{U_i\}$. Anyone in $\{U_i\}$ can verify if $c_j = \mathcal{G}_j(a_j)$ holds.

2: The set of actual signers $\{U_i\}$ agree on a random order $(1), (2), ..., (t)$ of their identities. According to the new order, each signer U_i is paired with its corresponding point $r_{(i)}$.

3: Each U_i randomly chooses its *voucher* α_i and computes

$$\begin{cases} v_{(i)} = H_2(\alpha_i, Sig_i(\alpha_i)) \\ w_{(i)} = H_1(v_{(i)}) \end{cases} \tag{3}$$

4: Each U_i broadcasts $v_{(i)}$, $w_{(i)}$, and $r_{(i)}$ to $\{U_i\}$. Anyone in $\mathcal{I}$ can verify if $w_{(i)} = H_1(v_{(i)})$ holds. Each signer keeps its α_i and $r_{(i)}$

5: The set of actual signers $\{U_i\}$ compute $h_0 = H_0(M, \mathcal{N}, t)$ and construct a polynomial f of degree n such that

$$\begin{cases} f(0) = h_0 \\ f(j) = c_j \\ f(r_{(i)}) = w_{(i)} \end{cases} \tag{4}$$

Then f is broadcast to $\{U_i\}$.

6: Each U_i computes $c_i = f(i)$ and $a_i = \mathcal{G}_i^{-1}(c_i)$.

7: The set of actual signers $\{U_i\}$ output the signature $\sigma = (M, \{U_m\}, t, f, \{a_1, .., a_n\}, \{v_{(1)}, ..v_{(t)}\})$

For any recipient of the message M and its signature σ, it can verify the signature by the following algorithm.

Signature Verification

1: Compute $h_0 = H_0(M, \{U_m\}, t)$, r_k, and $w_k = H_1(v_k)$ for $0 < k \leq t$.

2: Check the following:

$$\begin{cases} f(0) = h_0 = H_0(M, \{U_m\}, t) \\ f(m) = c_m = \mathcal{G}_m(a_m) \\ f(r_k) = w_k \quad 0 < k \leq t \end{cases} \tag{5}$$

If all conditions hold, accept, otherwise reject.

If the actual signer U_s is willing to prove to the recipient V that he is one of the actual signers, they run the following algorithm:

Signer Confession

1: V verifies the validity of the signature σ.

2: U_s uncovers his *voucher* α_s and $Sig_s(\alpha_s)$ to V, and tells V its corresponding $r_{(s)}$.

3: V verifies the signature $Sig_s(\alpha_s)$.

4: V accepts U_s as an actual signer if the all following conditions hold, rejects otherwise:

$$\begin{cases} v_{(s)} = H_2(\alpha_s, Sig_s(\alpha_s)) \\ w_{(s)} = H_1(v_{(s)}) \\ f(r_{(s)}) = w_{(s)} \end{cases} \tag{6}$$

Signer Denouncement

The property of denouncibility is optional. Suppose that the actual signer U_A has agreed to give U_B the authority to denounce itself, U_A sends its *voucher* α_A and $Sig_A(\alpha_A)$ to U_B. Then U_B can convince anyone U_A was involved in signing as in **Signer Confession**. Note that U_B need not to be one of the actual signers. We do not discuss the possibility that U_B might disclose α_A and $Sig_A(\alpha_A)$ to anyone else since we assume that U_A trust U_B. The individual signature $Sig_i(\alpha_i)$ of the *voucher* α_i is used for denouncement. If denouncibility is not necessary, $Sig_i(\alpha_i)$ can be omitted.

C. Scenario II

In this scenario, $t'(t' \leq t)$ or more members out of the t actual signers can jointly confess their involvement. However, since the confession can not be done by any individual, we do not need multiple r_k for each actual signer. Instead, a single r_s is used for confession. A trivial choice for r_s is set to be r_1.

Signature Generation

1: The same as the Step. 1 in *Scenario I.*
2: The set of actual signers $\{U_i\}$ compute the $r_s (= r_1)$.
3: Each U_i generates its *voucher* α_i and computes $\beta_i = H_3(\alpha_i, Sig_i\{\alpha_i\})$ then broadcasts β_i to $\{U_i\}$.
4: The set of actual signers $\{U_i\}$ runs the DKG protocol with input β_i and outputs the public key $v_s = g^d$. Note that β_i is kept secret by U_i.
5: The set of actual signers $\{U_i\}$ compute $h_0 = H_0(M, \{U_m\}, t, t')$ and $w_s = H_4(v_s)$ then construct a polynomial f of degree $n - t + 1$ such that

$$\begin{cases} f(0) = h_0 \\ f(j) = c_j \\ f(r_s) = w_s \end{cases} \tag{7}$$

Then f is broadcast to $\{U_i\}$.
6: Each U_i computes $c_i = f(i)$ and $a_i = \mathcal{G}_i^{-1}(c_i)$.
7: The set of actual signers $\{U_i\}$ output the signature $\sigma = (M, \{U_m\}, t, t', f, f', \{a_1, .., a_n\}, v_s)$

The verification algorithm is similar to the one in scenario I with minor difference.

Signature Verification

1: Compute $h_0 = H_0(M, \{U_m\}, t, t')$, r_1, and $w_s = H_4(v_s)$
2: Check the following:

$$\begin{cases} f(0) = h_0 = H_0(M, \{U_m\}, t, t') \\ f(m) = c_m = \mathcal{G}_m(a_m) \\ f(r_s) = w_s \end{cases} \tag{8}$$

If all condition hold then accept, otherwise reject .

If part of the original actual signers $\{U_{i'}\}$ ($\{U_{i'}\} \subseteq \{U_i\}$) is willing to prove to the recipient V that they arethe actual signers, they can run the following alogorithm:

Signers Threshold Confession

1: V verifies the validity of the signature σ.
2: Each $U_{i'}$ opens his *voucher* $\alpha_{i'}$ and $Sig_{i'}(\alpha_{i'})$ to V, and tells V its corresponding $r_{(i')}$.
3: V verifies each signature $Sig_{i'}(\alpha_{i'})$
4: V computes $\beta_{i'} = H_3(\alpha_{i'}, Sig_{i'}(\alpha_{i'}))$ and reconstructs the secret d by

$$d = f'(0) = \sum_{i \in \mathcal{I}'} \lambda_{i'}(0) \beta_{i'}. \tag{9}$$

5: V checks if $v_s = g^d$ holds. If it is true, V accepts that they are the actual signers.

IV. Realization Issues

We provide an example modified from [8], [10] to show how the three properties we proposed can be straightforward added to most ring signature schemes. We implement a scenario II scheme based on [8], [10]. Then we show an alternative of hash function we used to manipulate *vouchers*.

A. Core Threshold Ring Signature

The construction follows the ideas in [8], [10] which considered three possibilities: RSA-based, DL-based and ID-based. If signer U_i has a RSA-based keys pair, then its public key is (e_i, N_i) and private key is d_i where N_i is a product of two equal-length prime numbers and $e_i d_i \equiv 1 \quad mod \quad \phi(N_i)$. There exists a public hash function $\hat{H}_i : \{0,1\}^* \rightarrow Z^*_{N_i}$. If signer U_i has a DL-based key pair, then its public key is (p_i, q_i, g_i, y_i) and private key is $x_i \in Z^*_{q_i}$ where p_i, q_i are primes, $q_i | (p_i - 1)$, $g_i \in Z_{p_i}$. Finally, if U_i has a ID-based keys, this mean that there exist two cyclic group $G_{1,i}$ and $G_{2,i}$ of order q_i. The bilinear pairing is given as $\hat{e}_i : G_{1,i} \times G_{1,i} \rightarrow G_{2,i}$ and a public hash function $\hat{H}_i : \{0,1\}^* \rightarrow G_{1,i} - \{0\}$. U_i is under the control of a master entity whose master key is $x_i \in Z_{q_i}$ and public key is $Y_i = x_i P \in G_{1,i}$. Thus the public key of U_i is $PK_i = \hat{H}_i(U_i)$ and private key is $SK_i = x_i PK_i$.

For lake of space we only construct the scheme for scenario II, the construction for scenario I is similar.

Signature Generation

1: Similarly, but now $\mathcal{G}_j$ denotes the following operations: (1)For U_j, it computes

$$z_j = \begin{cases} g_j^{a_j} y_j^{c_j} \quad mod \quad p_j & \text{(DL-based)} \\ \hat{H}_j(c_j) + a_j^{e_j} \quad mod \quad N_j & \text{(RSA-based)} \\ e_j(a_j, P_j) \cdot e_j(Y_j, c_j PK_j) & \text{(ID-Based)} \end{cases} \tag{10}$$

where a_j is randomly chosen from $\{0,1\}^*$, Z_{q_j} or $G_{1,i}$, respectively, and c_j is picked from $\{0,1\}^l$.

(2)And for $\{U_i\}$, it computes

$$z_i = \begin{cases} g_i^{r_i} \quad mod \quad p_i & \text{(DL-based)} \\ r_i & \text{(RSA-based)} \\ e_i(T_i, P_i) & \text{(ID-Based)} \end{cases} \tag{11}$$

where r_i is randomly chosen from Z_{q_i} and T_i from $G_{1,i}$.

2: The set of actual signers $\{U_i\}$ compute the $r_s(= r_1)$.

3: Each U_i generates its *voucher* α_i and computes $\beta_i = H_3(\alpha_i, Sig_i\{\alpha_i\})$ then broadcasts β_i to $\{U_i\}$.

4: The set of actual signers $\{U_i\}$ run the DKG protocol with input β_i then outputs the public key $v_s = g^d$. Note that β_i is kept secret by U_i.

5: The set of actual signers $\{U_i\}$ compute $h_0 = H_0(M, \mathcal{N}, t, t')$ and $w_s = H_1(v_s)$ then construct a polynomial f of degree $n - t + 1$ such that

$$\begin{cases} f(0) = h_0 \\ f(j) = c_j \\ f(r_s) = w_s \end{cases} \tag{12}$$

Then f is broadcast to $\{U_i\}$.

6: Each U_i computes $c_i = f(i)$ but here the $\mathcal{G}_i^{-1}$ denotes the following operations:

$$a_i = \begin{cases} r_i - c_i x_i \quad mod \quad p_i & \text{(DL-based)} \\ (r_i - \hat{H}_i(c_i))^{d_i} \quad mod \quad N_i & \text{(RSA-based)} \\ T_i - c_i SK_i & \text{(ID-Based)} \end{cases} \tag{13}$$

7: The set of actual signers $\{U_i\}$ output the signature $\sigma = (M, \{U_i\}, t, t', f, f', \{a_1, .., a_n\}, v_s)$

Signature Verification

1: Compute $h_0 = H_0(M, \{U_i\}, t, t')$, r_1, and $w_s = H_4(v_s)$

2: Check the following:

$$\begin{aligned} & f(0) = h_0 = H_0(M, \{U_i\}, t, t') \\ & f(m) = c_m \\ & z_m = \begin{cases} g_m^{a_m} y_m^{c_m} \quad mod \quad p_m & \text{(DL-based)} \\ \hat{H}_m(c_m) + a_m^{e_m} \quad mod \quad N_m & \text{(RSA-based)} \\ e_m(a_m, P_m) \cdot e_m(Y_m, c_m PK_m) & \text{(ID-Based)} \end{cases} \\ & f(r_s) = w_s \end{aligned} \tag{14}$$

If all conditions hold then accept, reject otherwise.

B. Voucher Manipulation

We simply use hash functions H_1, H_2 to manipulate *voucher*. The idea of hash functions suggests the abstraction of the mechanism when dealing with voucher. The basic principle is that the vouchers are generated by some one-way function even for the actual signers may not able to reverse. From our view in [18], the part of signature $\{a, b, R, \epsilon\}$ has the same functionality as our *voucher* and the authors manipulated these tuple based on discrete-log problem. But in [18], the actual signer have to run an interactive proof with the verifier to prove its involvement. Actually many mechanisms possessing the property of one-way can replace the hash functions we used in manipulating the *vouchers*.

V. Security

Theorem 1 (Existential Unforgeability): Provided that each public key specifies a trapdoor one-way permutation, both our confessible threshold ring signature schemes are existentially unforgeable against adaptive chosen message attacks in the random oracle model [1].

Proof. Intuitively, the *vouchers*' roles in both our schemes are like additional points similar to h_0. Except for increasing the degree of the polynomial f, the core ring signature schemes remain the same. Since r_k are deterministic, it is impossible for the adversary to compute v_k (v_s in scenario II) such that $w_k = H_1(v_k)$ ($f(r_s) = w_s = H_4(v_s)$ in scenario II) in random oracle model. Provided that the core threshold ring signature scheme we use is existential unforgeable, for example in [5], unforgeability property is a matter of course. For lake of space we omit the tedious proof. In [5], [7], [8], [10], the authors had provide rigorous proofs of the unforgeability of ring signatures. Following the proof in Sec. 5 of [10] and Forking lemma in Sec 3 of [7], our claim is straightforward.

Theorem 2 (Signer Ambiguity): Provided that each public key specifies a trapdoor one-way permutation, both our confessible threshold ring signature schemes satisfy the property of unconditional signer ambiguity under the assumption that discrete logarithm problem is hard in the random oracle model.

Proof. Since for $\{U_m\}$, $\{a_m\}$ and all *vouchers* are randomly chosen, so $\{c_j\}$ and $\{v_{i'}\}$ (v_s in scenario II) are uniformly distributed. The polynomial f can be considered as a function chosen randomly from the collection of all polynomial with degree n ($n - t + 1$ in scenario II). Hence $\{c_i\}$ and then $\{a_i\}$ are also uniformly distributed. Briefly, for any message M and $\{U_m\}$, the distribution of σ are independent and uniformly distributed no matter which t actual signers are. So we conclude that even an adversary with all private keys of $\{U_m\}$ and unbounded computing resources has no advantage in identifying any actual signers.

VI. Conclusion

We presented two generic threshold ring signature schemes which satisfy the properties *confessibility* and *denouncibility*, or *threshold-confessibility* in two scenarios. Due to the anonymity, ring signatures have many applications in electronic commerce such as e-lotteries, e-voting, e-cash, etc. Also ring signatures solve many problems in ad-hoc and sensor networks or peer-to-peer environments. With the property of confessibility, our schemes derive many interesting applications. Future researches include extending the *threshold-confessibility* to general access structure or minimizing the signature size due to confessibility or denouncibility.

References

[1] M. Bellare and P. Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *ACM Conference on Computer and Communications Security*, pages 62–73, 1993.

[2] E. Bresson, J. Stern, and M. Szydlo. Threshold ring signatures and applications to ad-hoc groups. In *CRYPTO*, pages 465–480, 2002.

[3] T. Cao, D. Lin, and R. Xue. Improved ring authenticated encryption scheme. In *JICC*, 2004.

[4] T. Cao, D. Lin, and R. Xue. Id-based ring authenticated encryption. In *AINA*, pages 591–596, 2005.

[5] S. S. M. Chow, L. C. K. Hui, and S.-M. Yiu. Identity based threshold ring signature. In *ICISC*, pages 218–232, 2004.

[6] R. Gennaro, S. Jarecki, H. Krawczyk, and T. Rabin. Secure distributed key generation for discrete-log based cryptosystems. In *EUROCRYPT*, 1999.

[7] J. Herranz and G. Sáez. Forking lemmas for ring signature schemes. In *INDOCRYPT*, pages 266–279, 2003.

[8] J. Herranz and G. Sa'ez. Distributed ring signatures for identity-based scenarios. Cryptology ePrint Archive: Report 2004/377, 2004.

[9] J. Herranz and G. Sáez. Ring signature schemes for general ad-hoc access structures. In *ESAS*, pages 54–65, 2004.

[10] J. K. Liu, V. K. Wei, and D. S. Wong. A separable threshold ring signature scheme. In *ICISC*, pages 12–26, 2003.

[11] J. K. Liu, V. K. Wei, and D. S. Wong. Linkable spontaneous anonymous group signature for ad hoc groups (extended abstract). In *ACISP*, pages 325–335, 2004.

[12] J. Lv, K. Ren, X. Chen, and K. Kim. Ring authenticated encryption: A new type of authenticated encryption. In *SCIS*, 2004.

[13] J. Lv and X. Wong. Verifiable ring signature. In *DMS*, 2003.

[14] T. P. Pedersen. A threshold cryptosystem without a trusted party (extended abstract). In *EUROCRYPT*, pages 522–526, 1991.

[15] R. L. Rivest, A. Shamir, and Y. Tauman. How to leak a secret. In *ASIACRYPT*, pages 552–565, 2001.

[16] R. L. Rivest, A. Shamir, and Y. Tauman. How to leak a secret: Theory and applications of ring signatures. In *Essays in Memory of Shimon Even*, pages 164–186, 2006.

[17] P. P. Tsang, V. K. Wei, T. K. Chan, M. H. Au, J. K. Liu, and D. S. Wong. Separable linkable threshold ring signatures. In *INDOCRYPT*, pages 384–398, 2004.

[18] C. H. Wang and C. Y. Liu. A new ring signature scheme with signer-admission property. *Information Sciences (accepted).*, 2006.