# A Ring Signature Scheme with Strong Designated Verifiers to Provide Signer Anonymity

Shin-Jia Hwang
Department of Computer Science and Information Engineering,Tamkang University, Tamsui, Taipei Hsien, 251, Taiwan, R.O.C.
sjhwang@mail.tku.edu.tw

Kai-Lung Cheng
Department of Computer Science and Information Engineering,Tamkang University, Tamsui, Taipei Hsien, 251, Taiwan, R.O.C.
696410546@s96.tku.edu.tw

***Abstract*—To provide more anonymous protection for the actual signer, the first ring signatures with strong designated verifiers is proposed in 2007. In the ring signature scheme with strong designated verifiers, any member in the ad-hoc ring is able to generate ring signatures, so ring signatures provide signer ambiguity to protect the actual signer's anonymity. Moreover, only the designated verifier is convinced that the actual signer is one of the ring members. The other only guesses the actual signer may be one of the ring members and the designated verifier. If the other one guesses the actual signer being not only the ring members but also any possible ones, the scheme with strong designated verifiers provides the maximum anonymity protection for the actual signer. In order to convince the designated verifier, the designated verifier is still sure that the actual signer is one of the ring members. Therefore, the first ring signature scheme with strong designated verifiers is proposed to provide signer anonymity. In our scheme, the signer admission is also provided for the actual signer to prove who the actual signer is. The provable security analysis is provided to show that our scheme satisfies the ring signatures' correctness and unforgeability.**

***Index Terms*—Ring signature, designated verifier, anonymity, random oracle model.**

## I. INTRODUCTION

The concept of digital signature scheme is first proposed by Diffie and Hellman in 1976 [4]. In a digital signature scheme, a signer can transmit a message to the receiver. Then the receiver can authenticate the signature and message that the signature and message are really made by the signer. In general, a digital signature scheme satisfies these properties: soundness, completeness, and unforgeability [6].

However, the signer wants to keep his privacy without disclosing his true identity in some situation. So, Chaum and van Heyst proposed group signature schemes [3] to preserve signer's privacy. In the group signature scheme, a group manager is responsible to administrate the set of group members, to generate the public key for each group member, and to disclose the identity of the actual signer. The group signatures generated by each group member can be validated without knowing who the member is. To solving disputes, the group manager is also responsible to disclose the actual signer of some group signature. To protect the member's privacy, a group signature scheme satisfies two properties: signer ambiguity and signatures unlinkability [6].

Though group signature schemes preserve signers' privacy, the signer's privacy protection relies on the group manager's trusty. If the group manager is not trustworthy, then the signer's privacy is not protected at all. Moreover, in many situations, there is no group frame for some signers to generate anonymous signatures.

To solve this problem, in 2001, Rivest, Shamir, and Tauman proposed the concept of ring signature schemes [16]. In the ring signature scheme, there is no group frame, so a group manager is not needed. The actual signer arbitrarily specifies some innocent signers and the actual signer himself/herself to from an arbitrary set that is called a

ring. Then the actual signer generates a ring signature in ambiguous way that each member in the ring is able to generate the ring signature. To protect the actual signer's privacy, a ring signature scheme has to provide signer ambiguity at least. So the verifier only knows that the actual signer is somebody in the ring, but cannot know who the actual signer is.

An example is used to illustrate the application of ring signature schemes. For example, John is an employee of Allen's company, and he discovered that Allen will hollow out company's assets. John wants to send witness to a journalist without leaking his identity. So John chooses Allen's friends, customers, and employees to construct the ring. Then John signs a ring signature to a journalist. After validating the ring signature, the journalist is convinced by that all the members in the ring are the persons being possible knowing witness. However, the journalist does not confirm who the actual signer is.

After the first ring signature scheme is proposed, some ring signature schemes are proposed. In Naor's deniable ring signature schemes [12], the verifier validates ring signatures but cannot convince a third party that these ring signatures were signed by one of the ring members in 2000. Zhang and Kim proposed their identity-based ring signature scheme [19] in 2002. In the same year, Abe et al. proposed their separable ring signature schemes [1] in which all participants can choose their keys with different signature generation algorithms adopting different signature types. In 2003, Lv and Wang proposed their verifiable ring signature schemes [11]. In the verifiable ring signature scheme, the actual signer can prove to a recipient that the signature was signed by him/her. In 2004, Liu et al. proposed the linkable ring signature schemes [10]. In a linkable ring signature scheme, any two signatures are signed by the same ring member can be linked together. Ren and Harn proposed generalized ring signature scheme [15] in 2008 based on the original ElGamal signature scheme. Moreover, Ren and Harn's scheme is secure against adaptive chosen-message attack [7].

Among those proposed ring signature schemes, ring signatures are validated by anyone. In the above example, after obtaining and verifying the ring signature, Allen also knows the actual signer is some ring member and may be John. In fact, the actual signer does not want that Allen knows this betrayal, and wishes only the journalist can verify ring signatures. Therefore, a ring signature scheme with strong designated verifiers is needed for actual signers.

The designated verifier signature scheme was first introduced by Jakobsson et al [8]. In Jakobsson et al's scheme, anyone can validate the signature since only the signer's public key and the designated public key are used to validate signatures with designated verifier. But only the designated verifier is convinced who the actual signer is. In order to protect the designated verifier's privacy, Jakobsson et al defined the concept of strong designated verifier that only the designated verifier can validate signatures to identify the signer.

Lee and Chang proposed the first ring signature scheme [9] with strong designated verifier in 2007. In their scheme, only the designated verifier can validate the ring signature and be convinced that only some ring member generates the ring signature. Moreover, the designated verifier is also able to generate ring signatures that are indistinguishable from the ring signatures generated by some ring member. Therefore, Lee and Chang's scheme satisfies 1/n signer ambiguity only for the designated verifier, where n is the number of ring members. However, in Lee and Chang's scheme, the designated verifier is also able to generate indistinguishable ring signatures with strong designated verifier from those ring signature generated by some ring member. Therefore, Lee and Chang's scheme provides 1/(n+1) signer ambiguity for the other. Wu and Li [18] proposed another ID-based ring signature scheme to provided verifier specification. In order to specify many verifiers, Zhang and Xie proposed the ring signature scheme with multi-designated verifiers [20] based on the hardness of the chosen-target-inverse-CDH problem.

Among those proposed schemes, the actual sign-

er's identity is at most protected just among the ring members and the designated verifier. However, the most identity protection of the actual signer is among all possible users in the cryptosystem. In other words, the most protection is the signer anonymity, that is 1/max. signer ambiguity, for anyone except the actual signer and the designated verifier, where the denominator of 1/max. means all possible ones. To convince the designated verifiers, the ring signatures are still 1/n signer ambiguity for the designated verifiers. Because the designated verifier is also able to generate ring signatures, the actual signer needs to admit that the ring signature is made by him. However, those proposed schemes with strong designated verifiers do not provide the admission way for actual signers.

To provide the best anonymous protection for the actual signer and the most convince for the designated verifiers, the first ring signature scheme with strong designated verifiers is proposed to provide signer anonymity except designated verifiers. Moreover, in our new scheme, the signer admission algorithm is provided for the actual signer to admit that the ring signature is made by him.

In the next session, the reviews of the generalized ring signature scheme and the promise of Schnorr signature scheme are given. In Session III, the formal definition and security model of our ring signature scheme with strong designated verifiers are given first. Then our concrete ring scheme with strong designated verifiers is proposed to satisfy correctness, strong designated verifiers, unforgeability, signer anonymity, and signer ambiguity properties in the same session. The security proofs of our concrete scheme are given in Session IV. The last session is our conclusions.

## II. REVIEW

### A. Schnorr Signature Scheme and Corresponding Promises

Schnorr signature scheme [17] is given before the description of the promise of Schnorr signatures [13]. Schnorr signature scheme is consisted of three algorithms: SETUP, SIGN, and VERIFY.

**SETUP:**

The input of SETUP algorithm is a security parameter $l$. On this security parameter $l$, SETUP algorithm generates some public system parameters p, q, and $\gamma$, where p and q are two large primes with $q|(p-1)$ and $\gamma \in Z_p^*$ is an element with order q. SETUP algorithm also publishes a one-way hash function $h: \{0,1\}^* \rightarrow Z_q^*$. For each user $U_i$, his/her private key is a random integer $S_i \in Z_q^*$ and the corresponding public key is $P_i = \gamma^{S_i} \bmod p$.

**SIGN:**

SIGN algorithm is used to generate a signature on some message $M_i$. The input of SIGN algorithm is $(M_i, S_i)$ while the output of SIGN algorithm is the signature $(\sigma_i, c_i)$ on the message $M_i$. The concrete SIGN algorithm is stated below. Choose a random number $k_i \in Z_q^*$, compute $c_i = h(\gamma^{k_i}, M_i)$, and find $\sigma_i$ such that $\sigma_i \equiv k_i - c_i S_i \pmod q$. Then the Schnorr signature on the message $M_i$ is $(\sigma_i, c_i)$.

**VERIFY:**

The input of VERIFY algorithm is the triple $(\sigma_i, c_i, P_i)$. This algorithm outputs "accept" if $c_i = h(\gamma^{\delta_i} P_i^{c_i} \bmod p, M_i)$ holds; otherwise, outputs "reject".

The following theorem shows that Schnorr signature scheme is secure against passive adversary.

**Theorem 1:** If discrete logarithm (DL) problem is hard, then Schnorr signature can be secure against passive adversary in the random oracle model [14].

For a valid Schnorr signature $(\sigma_i, c_i)$ on the message $m_i$, the promise of $(\sigma_i, c_i)$ is $(\Sigma_i, c_i)$, where $\Sigma_i = \gamma^{\delta_i} \bmod p$. The verification of the promise $(\Sigma_i, c_i)$ is the equation $c_i = h(\Sigma_i \times P_i^{c_i} \bmod p, M_i)$.

The promise of Schnorr signature is forgeable. By using the public key $P_i$, everyone can forge a valid promise of some Schnorr signature at will. On any message $M_i'$, anyone randomly chooses k' from $Z_q^*$, and computes $c_i' = h(\gamma^{k'}, M_i')$ and $\Sigma_i' = (\gamma^{k'}/P_i^{c_i'}) \bmod p$. The equation $c_i' = h(\Sigma_i' \times P_i^{c_i'} \bmod p, M_i')$ holds, so $(\Sigma_i', c_i')$ is a forged promise of

Schnorr signature on any message $M_i'$ without the private key $S_i$.

### B. Generalized Ring Signature Scheme based on ElGamal Signature Scheme

The generalized ring signature scheme [15] proposed by Ren and Harn is described after the description of the ElGamal Signature Scheme [5]. The ElGamal signature scheme is consisted of three algorithms: SETUP, SIGN, and VERIFY.

**SETUP**:

The input of SETUP algorithm is a security parameter *l*. On this security parameter, SETUP algorithm first generates the public system parameters p and $g \in Z_p$, where g is an element with order p−1 over $Z_p$. Each user $U_i$ randomly chooses his/her private key $S_i \in Z_p$ and computes the corresponding public key $P_i = g^{S_i} \bmod p$.

**SIGN**:

SIGN algorithm is used to generate a signature on a message $m_i \in Z_{(p-1)}^*$. The input of SIGN is $(m_i, p, g, P_i, S_i)$, where p and g are system parameters, $P_i$ is the signer's public key, and $S_i$ is the signer's private key. The concrete SIGN algorithm is stated here. Choose a random integer $k_i \in Z_{(p-1)}^*$ and Compute $\alpha_i = g^{k_i} \bmod p$ and $\beta_i = k_i^{-1}(m_i - S_i\alpha_i) \bmod (p-1)$. Then the ElGamal signature on message $m_i$ is $(\alpha_i, \beta_i)$.

**VERIFY**:

The input of VERIFY algorithm is the tuple $(m_i, (\alpha_i, \beta_i), p, g, P_i)$. The algorithm outputs "accept" if the signature $(\alpha_i, \beta_i)$ is valid; otherwise, the algorithm outputs "reject". The concrete VERIFY algorithm is the validation of the equation $g^{\alpha_i}\alpha_i^{\beta_i} \equiv P_i^{m_i} \pmod p$. If $g^{\alpha_i}\alpha_i^{\beta_i} \equiv P_i^{m_i} \pmod p$ holds, then outputs "accept"; otherwise, outputs "reject".

According to [5], ElGamal signature scheme is existential forgeable under no message attack. In [5], the two-parameter forgery is proposed to forge ElGamal signature without private keys. Given the public key $P_i$, the attacker first randomly chooses $a_i$ from $Z_{p-1}$ and $b_i$ from $Z_{p-1}^*$. Then the attacker computes $\alpha_i' = g^{a_i}P_i^{b_i} \bmod p$, $\beta_i' = -\alpha_i' b_i^{-1} \bmod (p-1)$, and $m_i' = a_i\beta_i' \bmod (p-1)$. Finally, $(\alpha_i', \beta_i')$ is a valid ElGamal signature on the message $m_i'$ because $P_i^{\alpha_i'}\alpha_i'^{\beta_i'} \equiv P_i^{\alpha_i'}(g^{a_i}P_i^{b_i})^{\beta_i'} \equiv (g^{S_i})^{\alpha_i'}g^{a_i\beta_i'}(g^{S_ib_i})^{\beta_i'} \equiv (g^{S_i\alpha_i'})\, g^{m_i'}(g^{-S_i\alpha_i'}) \equiv g^{m_i'} \pmod p$ holds. The notation $G(a_i, b_i, P_i) = (\alpha_i', \beta_i', m_i')$ denotes this two-parameter forgery.

Ren and Harn's generalized ring signature scheme is consisted of three algorithms: SETUP, R-SIGN and R-VERIFY. SETUP algorithm is the same as the SETUP algorithm in ElGamal signature scheme, so only R-SIGN and R-VERIFY algorithms are described. Suppose that some actual user wants to generate a ring signature on a message M. The actual signer first chooses the ring member set $\{U_0, U_1, U_2, \ldots, U_{n-1}\}$ including the actual signer. The notation R denotes the public key set $\{P_0, P_1, P_2, \ldots, P_{n-1}\}$, where $P_i$ is the public key of the user $U_i$. Without losing generality, suppose that the actual signer is $U_s$, where $0 \le s \le (n-1)$.

**R-SIGN**:

The input of R-SIGN algorithm is the tuple (M, R, $S_s$, s), where s specifies the sth member, $U_s$ is the actual signer and $S_s$ is the actual signer's private key. The algorithm is stated as follows:

1. Choose a random starting value v from $Z_p$.
2. Forge the ElGamal signature $(\alpha_i, \beta_i)$ on some message $m_i$ for each ring member by using the two-parameter forgery G, except the actual signer $U_s$.
   2-1. Choose two random integers $a_i \in Z_{p-1}$ and $b_i \in Z_{p-1}^*$ for i= 0, 1, 2, …, (n−1) and $i \neq s$.
   2-2. Forge the ElGamal signature $(\alpha_i, \beta_i)$ by performing $G(a_i, b_i, P_i) = (\alpha_i, \beta_i, m_i)$. After forging $(\alpha_i, \beta_i)$, the message $m_i$ is also determined.
3. Compute $v_{s+1 \bmod n} = h(M, v)$, and $v_{s+j+1 \bmod n} = h(M, m_{s+j \bmod n} \oplus v_{s+j \bmod n})$ for j= 1, 2, ..., n−1. To fill the gap between v and $v_s$, compute $m_s = v \oplus v_s$.
4. Generate $(\alpha_s, \beta_s)$ on the message $m_s$ by using the actual signer's private key $S_s$.
   4-1. Choose a random integer $k_s$ from $Z_p^*$.
   4-2. Compute $(\alpha_s, \beta_s) = (g^{k_s} \bmod p, k_s^{-1}(m_s -$

$S_s\alpha_s) \bmod (p-1))$.

5. The ring signature on the message M is define: $\varphi = (R; r, v_r; m_1, m_2, \ldots, m_n; (\alpha_0, \beta_0), (\alpha_1, \beta_1), \ldots, (\alpha_{n-1}, \beta_{n-1}))$, where r is an integer randomly chosen between 0 and n−1.

**R-VERIFY**:

The input of R-VERIFY algorithm is the tuple (M, φ, R). This algorithm outputs "accept" if the signature φ is valid; otherwise, this algorithm outputs "reject". The concrete R-VERIFY algorithm is given in the following.

1. Verify equations $g^{m_i} = P_i^{\alpha_i}\alpha_i^{\beta_i} \bmod p$ for i= 0, 1, 2,…, (n−1). If any equation does not hold, outputs "reject" and stop.
2. Verify the equation $v_r = h(M, m_{r+(n-1) \bmod n} \oplus h(M, m_{r+(n-2) \bmod n} \oplus h(M, \ldots \oplus h(M, m_{r+1 \bmod n} \oplus h(M, m_{r \bmod n} \oplus v_{r \bmod n}))\ldots)))$. If this equation holds, then outputs "accept"; otherwise, outputs "reject".

Instead of using the primitive root as the parameter g, the generator with order q can be used as the parameter g for the ElGamal signature scheme. The existential forgery attack is still work for the ElGamal signature scheme using the generator g with order q. Here and after, the notation $G(a_i, b_i, P_i) = (\alpha_i', \beta_i', m_i')$ is redefined as the two-parameter forgery for the ElGamal signature scheme using the generator g with order q. The computational cost of $(\alpha_i', \beta_i')$ is about 1.16 modular exponentiations for $g^{a_i}P_i^{b_i} \bmod p$ and one modular inverse for $b_i^{-1} \bmod q$, where the double modular exponentiation $g^{a_i}P_i^{b_i} \bmod p$ is estimated by 1.16 modular exponentiations [2].

### C. Underlying Hard Problems

Since our proposed scheme is based on the hardness of DDHP, the problem is described below.

**Decision Diffie-Hellman Problem (DDHP)**

Let p, q be two large primes such that q|(p-1). Let g be an element with order q in $Z_p^*$. Given $g^a \bmod p$, $g^b \bmod p$, and $g^c \bmod p$ (where $a,b,c \in Z_q^*$ and be unknown), determine whether or not $g^c \equiv g^{ab} \pmod p$.

**DDHP Assumption**

There is no algorithm can solve DDHP in polynomial time with at least probability ε, where ε is negligible.

## III. Our Ring Signature Scheme with Strong Designated Verifiers

### A. Formal Definition of Our Scheme

Our ring signature scheme with strong designated verifiers is consisted of four algorithms: Setup, R-Sign, R-Ver and Admission.

**Setup(*l*)**:

On this security parameter *l*, the Setup algorithm first generates the public system parameters and public functions. This algorithm also generates the public key $P_i$ and private key $S_i$ for each user $U_i$.

**R-Sign(M, $P_1$, $P_2$, …, $P_n$, s, $S_s$, $P_v$)**:

Given a message M, the public keys $P_1$, $P_2$, …, $P_n$ of the n ring members, the actual signer $U_s$'s private key $S_s$, and the designated verifier's public key $P_v$, then Algorithm R-Sign produces a ring signature δ which contains the set of promises X on the message M. Algorithm R-Sign also produces a secret sender's evidence $\sigma_s$ which is used to convert the actual signer's promise containing in X to the original Schnorr signature.

**R-Ver(M, δ, $P_1$, $P_2$, …, $P_n$, $S_v$)**:

On the input consisting of the message M, a ring signature δ, the public keys $P_1$, $P_2$, …, $P_n$ of the n ring members, and the designated verifier's private key $S_v$, then Algorithm R-Ver determines whether or not (M, δ) is a valid ring signature.

**Admission(δ, $\sigma_s$, g, p):**

Given the ring signature δ and the evidence $\sigma_s$, if the one of promise set X in the ring signature δ is really constructed by $\sigma_s$, then outputs "accept"; otherwise, outputs "reject".

### B. Formal Security Model

A ring signature scheme should satisfy the following security properties. These security properties are defined below.

**Correctness:** If a ring signature is generated by R-Sign algorithm, then inputs to R-Ver algorithm always outputs "accept".

**Strong Designated Verifiers:** Only the designated verifier can validate ring signatures.

**Unforgeability:** R-Sign algorithm is existentially unforgeable against adaptive chosen message attack if any probabilistic polynomial time adversary *A* wins the follow game with non-negligible probability.

**GAME: (Existential Forgery Game by Adaptive Chosen Message Attacks)**

Let $L_{(init)}= \{P_1, P_2, \cdots, P_n\}$, $L_i= L_{(init)} \cup \{P_0\}$, where $P_0$ is the virtual user's public key. The adversary *A* makes the hash and signing queries polynomial-bounded times in polynomial-bounded order.

**Hash Query**: The adversary *A* sends the query to obtain the corresponding hash value h(x).

**Signing Query**: *A* sends the query $Q_i= (L_i, M_i)$ to the ring signing oracle *SO*, then *SO* returns $\delta_i$ being always accepted by Algorithm R-Ver if $L_i$ is a legal set consisting of legal users. Otherwise, *SO* returns the error message.

With the help of collecting $(L_i, M_i, \delta_i)$s, *A* outputs a forged ring signature $(L^*, M^*, \delta^*)$. Let $\{(L_i, M_i, \delta_i)\}$ denote the history of conversation between *SO* and *A*. The adversary *A* wins the game if $(L^*, M^*, \delta^*) \notin \{(L_i, M_i, \delta_i)\}$, $L^*$ is the set consisting of all legal users, and R-Ver$(L^*, M^*, \delta^*)$ always outputs "accept".

**Signer Anonymity:** It is unconditionally impossible to determine who produced ring signatures, even through the signer may be not in the group.

**Signer Ambiguity:** It is unconditionally impossible to determine which member produced given collection of signatures.

**C. Our Concrete Scheme**

Our concrete scheme is described by the four algorithms. In the following, the four algorithms are stated one by one.

**Setup(*l*)**:

The input of Setup algorithm is a security parameter *l*. On this security parameter, Setup algorithm first generates the public system parameters p, q, and g, where p and q are two large primes with $q|(p-1)$ and $g \in Z_p^*$ is an element with order q. Setup algorithm also publishes a one-way hash function $h: \{0, 1\}^* \rightarrow Z_q^*$, and makes a ring include n members $\{U_1, U_2, U_3, \ldots, U_n\}$. Each user $U_i$'s private key is a random integer $S_i \in Z_q^*$ and the corresponding public key is $P_i= g^{S_i} \bmod p$.

**R-Sign(M, $P_1$, $P_2$, …, $P_n$, s, $S_s$, $P_v$)**:

The actual signer performs the ring signature generation algorithm to generate the ring signature $\delta$ and the secret evidence $\sigma_s$ on the message M for the designated verifier is the user $U_v$. To be hidden among n possible signers, the actual signer randomly constructs the ring $\{U_1, U_2, \ldots, U_n\}$. Without losing generality, suppose that $U_s$ is the actual signer, where $1 \le s \le n$.

**Step 1**: Randomly choose $k \in Z_q^*$, and compute $P_0= (P_v)^k \bmod p$ and $W= g^k \bmod p$.

**Step 2**: Randomly choose $a_i \in Z_q$ and $b_i \in Z_q^*$, and forge the ElGamal signature $(\alpha_i, \beta_i)$ on the randomly message $m_i$ by computing $(\alpha_i, \beta_i, m_i)= G(a_i, b_i, P_i)$ for i= 0, 1, 2, …, n and $i \neq s$.

**Step 3**: Randomly choose $k_i' \in Z_q^*$, and compute $c_i= h(g^{k_i'} \bmod p, M)$ and $\Sigma_i= (g^{k_i'} P^{-c_i}) \bmod p$, for i=1, 2, …, n and $i \neq s$.

**Step 4**: Randomly choose $k_s' \in Z_q^*$, compute $c_s= h(g^{k_s'} \bmod p, M)$, and find $\sigma_s$ such that $\sigma_s \equiv k_s' - c_s S_s \pmod q$. Construct $(\Sigma_s, c_s)= (g^{\sigma_s} \bmod p, c_s)$ and keep the evidence $\sigma_s$ in secrecy.

**Step 5**: Construct $X= \{(\Sigma_1, c_1), (\Sigma_2, c_2), (\Sigma_3, c_3), \ldots, (\Sigma_n, c_n)\}$, and compute $D= h(M, P_0, P_1, P_2, \ldots, P_n, X, W)$.

**Step 6**: Randomly choose $V \in Z_q^*$,

$V_{s+1 \pmod{n+1}} = h(D, V)$,

$V_{s+2 \pmod{n+1}} = h(D, m_{s+1 \pmod{n+1}} \oplus V_{s+1 \pmod{n+1}})$

…

$V_{s+n \pmod{n+1}} = h(D, m_{s+n-1 \pmod{n+1}} \oplus V_{s+n-1 \pmod{n+1}})$,

$V_s = h(D, m_{s+n \pmod{n+1}} \oplus V_{s+n \pmod{n+1}})$.

**Step 7**: Compute $m_s = V_s \oplus V$ and the ElGamal signature $(\alpha_s, \beta_s)$ on the message $m_s$.

Finally, the ring signature on the message M is $\delta = (P_0, P_1, \ldots, P_n, r, V_r, m_0, m_1, \ldots, m_n, (\alpha_0, \beta_0), (\alpha_1, \beta_1), \ldots, (\alpha_n, \beta_n), X, W)$, where r is an integer randomly chosen between 0 and n.

**R-Ver($\delta$, $S_v$)**:

After receiving the ring signature $\delta = (P_0, P_1, \ldots, P_n, r, V_r, m_0, m_1, \ldots, m_n, (\alpha_0, \beta_0), (\alpha_1, \beta_1), \ldots, (\alpha_n, \beta_n), X, W)$ and the message M, the designated verifier $U_v$ verifies the ring signature $\delta$ in the following.

**Step 1:** Verify the trapdoor information $(\alpha_i, \beta_i)$ on the corresponding message $m_i$ by the equation $g^{m_i} = P_i^{\alpha_i} \alpha_i^{\beta_i} \bmod p$ for i= 0, 1,…, n. If some of the $(\alpha_i, \beta_i)$ are illegal, then stop.

**Step 2:** Compute $D = h(M, P_0, P_1, P_2, \ldots, P_n, X, W)$.

**Step 3:** Validate the ring signature $\delta$ by verifying the verification equation $V_r = h(D, m_{r+n \pmod{n+1}} \oplus h(D, m_{r+n-1 \pmod{n+1}} \oplus h(D, m_{r+n-2 \pmod{n+1}} \oplus \ldots \oplus h(D, m_{r+n-(n-1) \pmod{n+1}} \oplus h(D, m_r \oplus V_r)) \ldots)))$.

**Step 4:** Validate the correctness of $P_0$ and W by the equation $P_0 \equiv W^{S_v} \pmod p$.

**Step 5:** Verify the promise $(\Sigma_i, c_i)$ on the message M by the equation $c_i = h(\Sigma_i \times P_i^{c_i} \bmod p, M)$ for i= 1, 2,…, n.

If the ring signature $\delta$ can pass previous five steps, then outputs "accept"; otherwise, outputs "reject".

**Admission($\delta$, $\sigma_s$)**:

Suppose that the given ring signature $\delta = (P_0, P_1, \ldots, P_n, r, V_r, m_0, m_1, \ldots, m_n, (\alpha_0, \beta_0), (\alpha_1, \beta_1), \ldots, (\alpha_n, \beta_n), X, W)$ on some message M passes the verification of R-Ver. The actual signer $U_s$ provides the actual evidence $\sigma_s$ to guard against the forgery of the designated verifier. After getting the $\sigma_s$ from $U_s$, the witness or $U_v$ is able to verify $\sigma_s$ by the equation $g^{\sigma_s} \equiv \Sigma_s \pmod p$. If $g^{\sigma_s} \equiv \Sigma_s \pmod p$ holds, then the witness or $U_v$ accepts that $U_s$ is the actual signer.

## IV. Security Analysis

The security analysis of our ring signature scheme with strong designated verifiers is given by discussing five properties: correctness, strong designated verifiers, unforgeability, signer ambiguity, and signer anonymity.

### A. Correctness

**Theorem 2**: A ring signature generated by R-Sign algorithm must be accepted by R-Ver algorithm.

**Proof:** The three major verifications used in Algorithm R-Ver are used to verify the trapdoor information, the ring equation, and the correctness of $P_0$ and W. These three proofs are given one by one.

**1. Verify the trap-door information:** In our scheme, the trapdoor information $(\alpha_i, \beta_i)$ on the corresponding message $m_i$ is validated by $g^{m_i} \equiv g^{a_i \beta_i} \equiv P_i^{\alpha_i} \times g^{a_i \beta_i} \times P_i^{b_i \beta_i} \equiv P_i^{\alpha_i} \times (g^{a_i} \times P_i^{b_i})^{\beta_i} \equiv P_i^{\alpha_i} \alpha_i^{\beta_i} \pmod p$ for i= 1, 2, …, n and $i \neq s$. Since the signature $(\alpha_s, \beta_s)$ is a legal ElGamal signature on the message $m_s$, $g^{m_s} \equiv P_s^{\alpha_s} \times \alpha_s^{\beta_s} \pmod p$ holds.

**2. Verify the ring equation:** Consider the first case that s>r. In order to validate the correctness of $(V_r, m_0, m_1, \ldots, m_n)$, the verifier performing the following computations.

$V_{r+1 \pmod{n+1}} = h(D, m_{r \pmod{n+1}} \oplus V_{r \pmod{n+1}})$

$V_{r+2 \pmod{n+1}} = h(D, m_{r+1 \pmod{n+1}} \oplus V_{r+1 \pmod{n+1}})$

…

$V_{s+1 \pmod{n+1}} = h(D, m_{s \pmod{n+1}} \oplus V_{s \pmod{n+1}}) = h(D, V)$

$V_{s+2 \pmod{n+1}} = h(D, m_{s+1 \pmod{n+1}} \oplus V_{s+1 \pmod{n+1}})$

….

$V_{r \pmod{n+1}} = h(D, m_{r-1 \pmod{n+1}} \oplus V_{r-1 \pmod{n+1}})$.

Since the second case that s<r is similar to the first case, by similar reasoning, $(V_r, m_0, m_1, \ldots, m_n)$ passes the verification of the ring equation.

**3. Validate the correctness of $P_0$ and W:** The correct pair $(P_0, W)$ should satisfy $P_0 \equiv (P_v)^k \equiv (g^{S_v})^k \equiv W^{S_v} \pmod{p}$.

### B. Strong Designated Verifiers

In order to specify the verifier, in our scheme, Lemma 1 shows that no one can verify the ring signature except the signer and the designated verifier.

**Lemma 1:**

Given $W= g^k \bmod p$ and $P_0= g^{kS_v} \bmod p$, except the signer and the verifier, no one can determine whether or not $P_0= (W)^{S_v} \bmod p= (g)^{kS_v} \bmod p$.

**Proof:** Given $W= g^k \bmod p$, $P_v$ and $P_0= g^{kS_v} \bmod p$, to determine whether or not $P_0= (W)^{S_v} \bmod p= (P_v)^k \bmod p= g^{kS_v} \bmod p$ is equivalent to the Decision Diffie-Hellman Problem (DDHP). Therefore, no one solve it efficiently.

### C. Unforgeability

The unforgeability proof of our ring signature scheme consists of two cases. One case is that the unforgeability for the designated verifier while another is that the unforgeability for anyone, expect the designated verifier.

**Case1**: The designated verifier is able to forge the ring signature δ, but cannot forge the evidence $\sigma_s$.

Consider a ring signature $\delta= (P_0, P_1,\ldots, P_n, r, V_r, m_0, m_1, \ldots, m_n, (\alpha_0, \beta_0), (\alpha_1, \beta_1), \ldots, (\alpha_n, \beta_n), X, W)$ on some message M. The designated verifier easily forges all components without knowing the evidence $\sigma_s$. The designated verifier first chooses the random integer k, and computes $P_0= (P_v)^k \bmod p$ and $W= g^k \bmod p$. Then the private key of $P_0= g^{kS_v} \bmod p$ is $S_0= k \times S_v \bmod q$. Since $S_v$ is the designated verifier's private key, only the designated verifier can find the value of $S_0$.

Then the receiver constructs the ring signature on the message M by the following way. According to the forgery in Session II-A, the promise set $X= \{(\Sigma_1 , c_1), (\Sigma_2 , c_2), (\Sigma_3 , c_3),\ldots, (\Sigma_n , c_n)\}$ is easily forged without knowing the evidence $\sigma_s$. Then the designated verifier forges the ElGamal signature $(\alpha_i, \beta_i)$ on the randomly generated message $m_i$ by computing $(\alpha_i, \beta_i, m_i)= G(a_i, b_i, P_i)$ for i= 1, 2, …, n. The designated verifier also chooses another two random integers V and W, and computes $D= h(M, P_0, P_1, P_2,\ldots, P_n, X, W)$. Then the designated verifier computes

$V_1 = h(D, V)$,

$V_2 = h(D, m_1 \oplus V_1)$,

…

$V_n = h(D, m_{n-1} \oplus V_{n-1})$, and

$m_0 = V \oplus V_n$.

The ElGamal signature $(\alpha_0, \beta_0)$ on $m_0$ can be forged since the designated verifier knows the private key $S_0$. Finally, the ring signature $\delta= (P_0, P_1,\ldots, P_n, r, V_r, m_0, m_1, \ldots, m_n, (\alpha_0, \beta_0), (\alpha_1, \beta_1), \ldots, (\alpha_n, \beta_n), X, W)$ on some message M is forged without generating the evidence $\sigma_s$.

If the designated verifier can forge the ring signature δ on some message with knowing the evidence, then the forgery algorithm can be used to forge Schnorr signature on any message.

**Lemma 2**:

The unforgeability of the ring signature δ with the evidence $\sigma_s$ by the designated verifier is based on the unforgeability of Schnorr signature scheme.

**Proof**: This proof is obvious. Suppose that an algorithm Ring-FV can be used to forge the ring signature with the evidence by the designated verifier. Then the algorithm Ring-FV can be used to forge a Schnorr signature scheme on the public key P and message M'. Let $P_s= P$ and M= M'. Then the forged ring signature $\delta= (P_0, P_1,\ldots, P_n, r, V_r, m_0, m_1, \ldots, m_n, (\alpha_0, \beta_0), (\alpha_1, \beta_1), \ldots, (\alpha_n, \beta_n), X, W)$ with the evidence $\sigma_s$, where the promise set $X= \{(\Sigma_1 , c_1), (\Sigma_2 , c_2), (\Sigma_3 , c_3),\ldots, (\Sigma_n , c_n)\}$. With the help of $(\Sigma_s, c_s)$ and $\sigma_s$, the Schnorr signature $(\sigma_s,$

$c_s$) on the message M' is forged successfully.

**Case2**: Except the designated verifier, anyone cannot forge the ring signature δ for the unforgeability of the ElGamal signature.

**Unforgeability-game:** Assume the T generates the system parameters and sends them to *A*. To win the game, *A* performs the queries, H-oracle and Ring signing oracle below many times.

After a period of time, *A* outputs a tuple (M*, δ*), where M* was not queried for the signature as shown above. The adversary *A* wins the game if δ* is a valid ring signature on the message M*.

**H-oracle**

Assume that a list LH keeps track of answers to the hash queries on the H-oracle. To simulate the hash function H, T checks the list LH. If the query was made by the adversary *A* previously, T returns the same answer as that in LH. If the query is a new one, T randomly chooses a number and inserts it into LH as a new entry. The detailed method is described as follows. Since *A* is polynomial-time bounded, let the input is $x_i$ for the ith query. Note that the output of the H-oracle, $H_i$= $h(x_i)$. The new entry of LH for the ith query is the tuple of $\{x_i, H_i\}$. If the query is given from the ring signing oracle with designated input (D, m', V'), V, and $m_s$, then H-oracle returns the $h(D, m' \oplus V')= V \oplus m_s$.

**Ring Signing Oracle *SO***

**Step 1:** Randomly choose $k \in Z_q^*$ and compute $W = (g)^k \bmod p$ and $P_0 = P_v^k \bmod p$.

**Step 2:** **Randomly** choose $k_i' \in Z_q^*$, and compute $c_i= h(g^{k_i'}, M)$ and $\Sigma_i= (g^{k_i'} p^{-c_i}) \bmod p$, for i= 1, 2, …, n.

**Step 3:** Construct $X= \{(\Sigma_1, c_1), (\Sigma_2, c_2), (\Sigma_3, c_3),\ldots, (\Sigma_n, c_n)\}$ and $D = h(M, P_0, P_1, P_2,\ldots, P_n, X, W)$.

**Step 4:** Randomly choose $a_i \in Z_q$ and $b_i \in Z_q^*$, and forge the ElGamal signature $(\alpha_i, \beta_i)$ on the randomly generated message $m_i$ by computing $(\alpha_i, \beta_i, m_i)= G(a_i, b_i, P_i)$ for i= 0, 1, 2, …, n.

**Step 5:** Randomly choose $V \in Z_q^*$, and compute $V_{s+1 \,(\text{mod } n+1)}= h(D, V)$, $V_{s+2 \,(\text{mod } n+1)}= h(D, m_{s+1 \,(\text{mod } n+1)} \oplus V_{s+1 \,(\text{mod } n+1)})$, …, $V_{s+n \,(\text{mod } n+1)}= h(D, m_{s+n-1 \,(\text{mod } n+1)} \oplus V_{s+n-1 \,(\text{mod } n+1)})$.

**Step 6:** Find $h(D, m_{s+n \,(\text{mod } n+1)} \oplus V_{s+n \,(\text{mod } n+1)})= V_s= V \oplus m_s$ by sending hash query with the designated input $(D, m_{s+n \,(\text{mod } n+1)}, V_{s+n \,(\text{mod } n+1)})$, V, and $m_s$.

Ring signature $\delta= (P_0, P_1, P_2, \ldots, P_n, r, V_r, m_0, m_1, m_2,\ldots, m_n, (\alpha_0, \beta_0), (\alpha_1, \beta_1), (\alpha_2, \beta_2), \ldots, (\alpha_n, \beta_n), X, W)$, where r is an integer randomly chosen between 0 and n.

**Lemma 3**:

Our ring signature scheme with strong designated verifiers is secure against adaptive chosen message attack. In other words, if adversary *A* wins the unforgeability-game by forging the ring signature, then T can adopt *A* to forge an ElGamal signature on the designated message m′ without known the signer's private key $S_s$.

**Proof:** Assume that the adversary *A* wins the unforgeability-game by forging ring signatures. In the following, an algorithm T on the public key P and the message m' is designated to forge an ElGamal signature on m' without known the private key of P.

First of all, let $P_s$= P. The algorithm T randomly generates the message M, the public key $P_v$, and the public keys $P_i$ for i= 1, 2,…, n and i≠ s. Then $(P_1, P_2,\ldots, P_n)$, $P_v$, and the message M is the input of the adversary *A*. By utilizing the ring signing oracle *SO*, the adversary *A* is able to collect the number of the ring signatures on some messages chosen by *A*.

When the adversary *A* wants to forge the ring signature on the message M, then T controls H-oracle such that $V_s= h(D, m_{s+n(\text{mod } n+1)} \oplus V_{s+n(\text{mod } n+1)})= V \oplus m'$ for the designated input $(D, m_{s+n(\text{mod } n+1)}, V_{s+n(\text{mod } n+1)})$, V, and m' from the adversary *A*. After *A* querying H-oracle, *A* successfully forges the ring signature $(P_0, P_1, P_2,\ldots, P_n, r, V_r, m_0, m_1, m_2,\ldots, m_n, (\alpha_0, \beta_0), (\alpha_1, \beta_1), (\alpha_2, \beta_2),\ldots, (\alpha_n, \beta_n), X, W)$ on the message M. Since $m_s$= m', T obtains

the ElGamal signature ($\alpha_s$, $\beta_s$) on the designated message m' without the signer's private key $S_s$.

### D. Signer Ambiguity

Our scheme provides signer ambiguity for the designated verifier. At the starting point, since k is taken randomly from $Z_q^*$, so $P_0$ and W are distributed uniformly over $Z_p^*$. To generate ElGamal signatures, because $a_i$ randomly choose from $Z_q$ and $b_i$ randomly choose from $Z_q^*$, then the forged ElGamal signature ($\alpha_i$, $\beta_i$) and the corresponding message $m_i$ are also distributed uniformly. To generate the promises of Schnorr signature, since $k_i'$ and $k_s'$ are randomly chosen from $Z_q^*$, the promise of Schnorr signature ($\Sigma_i$, $c_i$) and X= {($\Sigma_1$ , $c_1$), ($\Sigma_2$ , $c_2$), ($\Sigma_3$ , $c_3$),…, ($\Sigma_n$ , $c_n$)} are distributed uniformly. Therefore, for fixed *l*, M, and the designated verifier, any $U_i$ in {$U_1$, $U_2$,…, $U_n$} has the same probability to generate the ring signature.

### E. Signer Anonymity

The signer anonymity is used to provide the privacy protection of the signer for anyone. By Lemma 1, only the designated verifier $U_v$ can verify $P_0 = W^{S_v} \bmod p$. Therefore, no one is able to validate the ring signature except the designated verifier. Lemma 4 shows that the ring signature is forgeable if the relation among $P_0$, $P_v$, and W is not validated.

**Lemma 4**:

Because nobody except the designated verifier $U_v$ can verify the relation among $P_0$, $P_v$, and W, anyone even is not the ring member can forge the ring signature on the message M with help of the non-existent public key $P_0$.

**Proof**: Anyone can forge the signature with the nonexistent signer $P_0$ by follow steps.

**Step 1**: Randomly choose $k \in Z_q^*$, and compute $W = (g)^k \bmod p$.

**Step 2**: Randomly choose $S_0 \in Z_q^*$, and compute $P_0 = g^{S_0} \bmod p$.

**Step 3**: Randomly choose $k_i' \in Z_q^*$, and compute $c_i = h(g^{k_i'}, m)$ and $\Sigma_i = (g^{k_i'} p^{-c_i}) \bmod p$, for i= 1, 2, …, n.

**Step 4**: Construct X= {($\Sigma_1$ , $c_1$), ($\Sigma_2$ , $c_2$), ($\Sigma_3$ , $c_3$),…, ($\Sigma_n$ , $c_n$)} and $D = h(M, P_0, P_1, P_2,\ldots, P_n, X, W)$.

**Step 5**: Randomly choose $V \in_R Z_q^*$

$V_{0+1 \,(\mathrm{mod}\, n+1)} = h(D, V)$,

$V_{0+2 \,(\mathrm{mod}\, n+1)} = h(D, m_{0+1 \,(\mathrm{mod}\, n+1)} \oplus V_{0+1 \,(\mathrm{mod}\, n+1)})$

…

$V_{0+n \,(\mathrm{mod}\, n+1)} = h(D, m_{0+n-1 \,(\mathrm{mod}\, n+1)} \oplus V_{0+n-1 \,(\mathrm{mod}\, n+1)})$,

$V_0 = h(D, m_{0+n \,(\mathrm{mod}\, n+1)} \oplus V_{0+n \,(\mathrm{mod}\, n+1)})$.

**Step 6**: Compute $m_0 = V_0 \oplus V$ and construct the ElGamal signature ($\alpha_0$, $\beta_0$) on the message $m_0$ by using the private key $S_0$.

Finally, the ring signature on the message M is $\delta = (P_0, P_1,\ldots, P_n, r, V_r, m_0, m_1, \ldots, m_n, (\alpha_0, \beta_0), (\alpha_1, \beta_1), \ldots, (\alpha_n, \beta_n), X, W)$, where r is an integer randomly chosen between 0 and n.

Consider the signer ambiguity from the viewpoint of the designated verifier $U_v$. First, because $P_0 = (P_v)^k \bmod p$, only the designated verifier $U_v$ can validate whether or not $P_0$ is forged. In other words, only $U_v$ confirms that the actual signer must be one of {$U_1$, $U_2$, $U_3$,…,$U_n$} because the private key $S_0$ of $P_0 = (P_v)^k \bmod p$ is unknown for anyone except $U_v$. Second, because $a_i$, $b_i$, and $m_i$ are all random chosen, so they are uniform distributed. The probability of guessing the actual signer is 1/n, so the signer ambiguity is satisfied.

Consider the signer anonymity from the viewpoint of the others except the designated verifier $U_v$. Anyone cannot verify the relation among $P_0$, $P_v$, and W, so nobody knows whether or not the ring signature is forged. Then the probability of guessing the actual signer is 1/max. because ring signatures may be forged by anyone instead of the n members $U_1$, $U_2$, $U_3$,…, and $U_n$. Therefore signer anonymity is satisfied.

## V. CONCLUSIONS

To provide the signer anonymity among all possible signers, the ring signature scheme with strong designated verifiers is proposed. In our scheme, since the actual signer is hidden among all signers for the other users except the designated verifier, the actual signer is protected by signer anonymity. Only the designated verifier can validate a ring signature with strong designated verifiers as an ordinary ring signature. So only the designated verifier is convinced that the actual signer is one member among the ring. For this signer ambiguity, the actual signer's identity is protected and the designated verifier is also convinced. Our scheme provides the best anonymous protection for the actual signer and the most convince for the designated verifiers. On the other hand, the two proposed ring signature schemes with strong designated verifiers only provide signer ambiguity protection to hide the actual signer's identity. Moreover, only our scheme provides the signer admission algorithm to enable the actual signer to show who the actual signer is. The security proof is provided to show that our scheme satisfies correctness, signer anonymity for any verifier except the designated verifier, signer ambiguity for designated verifiers, and unforgeability against adaptive chosen message attacks in the random oracle model. However, the other two proposed scheme do not provide the formal security proof. Our scheme with provable security is better than the other two proposed schemes because only our scheme provides signer anonymity for the others and signer admission.

## REFERENCE


[1] M. Abe, M. Ohkubo, and K. Suzuki, "1-out-of-n Signatures from a Variety of Keys," *Advances in Cryptology-ASIACRYPT 2002*, LNCS 2501, Berlin: Springer Verlag, 2002, pp.415-432.

[2] G. Ateniese, "Efficient Verifiable Encryption (and Fair Exchange) of Digital Signature," *Proceedings of 6th ACM Conference on Computer and Communications Security*, New York: ACM, 1999, pp. 138-146.

[3] D. Chaum and E. V. Heyst, "Group signatures," *Advances in Cryptology-Eurocrypt'91*, LNCS 547, New York: Springer Verlag, 1991, pp. 257-265.

[4] W. Diffie and M. Hellman, "New directions in cryptography," *IEEE Transactions on Information Theory*, Vol. 22, Issue 6, pp. 644-654, Nov. 1976.

[5] T. A. ElGamal, "A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms," *IEEE Transactions on Information Theory*, Vol. 31, No. 4, pp. 469-472, July. 1985.

[6] C. Gamage, B. Gras, B. Crispo, and A. S Tanenbaum, "An Identity-based Ring Signature Scheme with Enhanced Privacy," *Securecomm and Workshops*, Baltimore, MD, Aug. 28-Sep. 1, 2006, pp. 1-5.

[7] S. Goldwasser, S. Micali, and R. L. Rivest, "A Digital Signature Scheme Secure against Adaptive Chosen-Message Attacks," *SIAM Journal on Computing*, Vol. 17, Issue 2, pp. 281-308, 1988.

[8] M. Jakobsson, K. Sako and R. Impagliazzo, "Designated Verifier Proofs and Their Applications," *Advances in Cryptology-Eurocrypt'96*, LNCS 1070, Berlin: Springer Verlag, 1996, pp.143-154.

[9] J. S. Lee and J. H. Chang, "Strong Designated Verifier Ring Signature Scheme," *Innovations and Advanced Techniques in Computer and Information Sciences and Engineering*, Netherlands: Springer Verlag, 2007, pp.543-547.

[10] J. K. Liu, V. K. Wei, and D. S. Wong, "Linkable Spontaneous Anonymous Group Signature for Ad Hoc Groups," *Information Security and Privacy*, LNCS 3108, Berlin: Springer Verlag, 2004, pp. 325-335.

[11] J. Lv and X. Wang, "Verifiable Ring Signature," *Proceedings of DMS- The 9th International Conference on Distributed Multimedia*

*Systems*, Miami, Florida, USA, Sep. 2003, pp. 663-667.

[12] M. Naor, "Deniable Ring Authentication," *Advances in Cryptology-CRYPTO 2002*, LNCS 2442, Berlin: Springer Verlag, 2002, pp. 481-498.

[13] K. Nguyen, "Asymmetric Concurrent Signatures," *Information and Communications Security*, LNCS 3783, Berlin: Springer Verlag, 2005, pp. 181-193.

[14] D. Pointcheval and J. Stern, "Security Arguments for Digital Signatures and Blind Signatures," *Journal of Cryptology*, Vol. 13, No. 3, New York: Springer Verlag, pp. 361-396, Dec. 2000.

[15] J. Ren and L. Harn, "Generalized Ring Signature," *IEEE Transactions on Dependable and Secure Computing*, Vol. 5, Issue 3, pp. 155-163, July-Sept. 2008.

[16] R. L. Rivest, A. Shamir, and Y. Tauman, "How to Leak a Secret," *Advances in Cryptology-ASIACRYPT 2001*, LNCS 2248, Berlin: Springer Verlag, 2001, pp. 552-565.

[17] C. P. Schnorr, "Efficient Identification and Signatures for Smart Cards," *Advances in Cryptology-CRYPTO'89 Proceedings*, LNCS 435, Berlin: Springer Verlag, 1990, pp. 239-252.

[18] L. Wu and D. Li, "Strong Designated Verifier ID-Based Ring Signature Scheme," *Information Science and Engineering, 2008, ISISE'08, International Symposium,* Vol. 1, Shanghai, People's Republic of China, Dec. 20-22, 2008, pp.294-298.

[19] F. Zhang and K. Kim, "ID-Based Blind Signature and Ring Signature from Pairings," *Advances in Cryptology-ASIACRYPT 2002*, LNCS 2501, Berlin: Springer Verlag, 2002, pp. 533-547.

[20] J. Zhang and J. Xie, "A Novel Ring Signature Scheme with Multi-designated Verifiers," *IEEE International Conference on Cybernetics and Intelligent Systems*, Chengdu, People's Republic of China, Sep. 21-24, 2008, pp.873-877.