JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 30, 895-910 (2014)

DOI:10.1688/JISE.2014.30.3.20

A Strong Designated Verifier Ring Signcryption Scheme Providing Strongest Signers' Anonymity

SHIN-JIA HWANG AND JYONG-YE CHEN*
Department of Computer Science and Information Engineering
Tamkang University
Tamsui, New Taipei City, 251 Taiwan
*E-mail: {sjhwang@mail; *697410727@s97}.tku.edu.tw*

The strong designated verifier ring signature scheme provides signer anonymity to protect actual signer's identity. However, the message of the strong designated verifier ring signature may disclose some identity information related to the actual signer. To remove this flaw, this study proposes a novel strong designated verifier ring signcryption scheme. Compared with the Han *et al.* and Huang *et al.* ring signcryption schemes, the proposed scheme provides strongest signer anonymity to protect the signer identity. This scheme also provides signer admission to show who the actual signer is. Unlike some proposed schemes, which still suffer the message length restriction, this scheme is free from the message length restriction to provide message confidentiality.

Keywords: ring signcryption, signer anonymity, strong ring signature scheme with designated verifiers, signatures, encryption

1. INTRODUCTION

The ring signature scheme [1] provides one-out-of-*n* signer anonymity among an ad hoc group, which is called a ring, consisting of *n* members. In the ring signature scheme, although anyone can validate ring signatures, no one is able to identify who the actual signer is. For one-out-of-*n* signer anonymity, ring signature schemes have many practical applications such as news reports with anonymous signers and *e*-voting schemes.

After the first ring signature scheme was proposed, various types of ring signature schemes were proposed. The first ID-based ring signature scheme [2] based on the ID-based cryptosystem [3] was proposed to remove public key certificates. The linkable ring signature scheme [4] was proposed such that the linkable ring signatures generated by the same signer can be clustered. Because of its linkable property, the linkable ring signature scheme is useful for implementing the *e*-voting scheme. The (*t*, *n*) threshold ring signature scheme [5] generates the ring signatures by only the cooperation of t or more ring members. Based on the ElGamal signature scheme, the generalized ring signature scheme [6] was proposed to provide signer admission to prove who the actual signer is. However, the generalized ring signature scheme does not provide signer admission [7], and an improved scheme [7] was proposed to provide the signer admission.

In these schemes, anyone can validate ring signatures. In some applications, signers may want to specify their ring signature being validated only by some designated-verifier. For example, someone in the government wants to publish news to the designated reporter, such that no one can verify the news except the designated reporter. Although the designated-verifier is able to validate ring signatures, he/she still cannot identify the ac-

Received April 13, 2012; revised February 15, 2013; accepted June 11, 2013.
Communicated by Wen-Guey Tzeng.

tual signer.

For the designated-verifier requirement, the designated verifier ring signature scheme was proposed [8-10]. Based on the designated-verifier property, only the designated verifier is able to validate ring signatures. Among these schemes, only the Hwang and Cheng scheme [10] provides one-out-of-all signer anonymity for anyone, except the designated verifier. The other schemes provide only one-out-of-n signer anonymity for anyone. The one-out-of-all signer anonymity property means that the actual signer may be any legal user in the system. Therefore, Hwang and Cheng's scheme provides the strongest identity privacy protection, and also provides signer admission. Hwang and Chen [11] proposed an improvement to enhance the performance of Hwang and Cheng's scheme.

Schemes with designated verifiers do not protect message confidentiality. However, the message sent in plaintext may release some sensitive information. For example, a whistleblower in the government wants to send sensitive and secret news to a designated reporter. The whistleblower keeps not only his identity, but also the news, a secret from anyone except the designated reporter. Because the sensitive news may be known by only a few people, the sensitive news releases some useful identity information on the whistleblower. Thus, the message confidentiality is as important as the unforgeability and anonymity properties for the designated verifiers ring signature schemes.

Inspired by the signcryption schemes [12], (strong) designed verifiers ring signcryption schemes [13, 14] can provide confidentiality and ring signatures concurrently. However, some ring signcryption schemes [16-21] are insecure [15]. The ring signcryption schemes [13, 14] integrate ring signature schemes and symmetric cryptosystems to sign and encrypt messages anonymously and efficiently. Although ring-signcryptexts cannot be decrypted and verified, the actual signer is still only a ring member; therefore, these schemes only provide one-out-of-n signer anonymity.

To provide one-out-of-all signer anonymity, this study proposes a novel strong designated verifier ring signcryption scheme based on Hwang and Chen's strong designated verifier ring signature scheme. Section 2 details the proposed scheme. Section 3 provides the security proof for this scheme. Section 4 presents a comparison of ring signcryption schemes. Finally, Section 5 offers a conclusion.

2. STRONG DESIGNATED VERIFIER RING SIGNCRYPTION SCHEME

Our strong designated verifier ring signcryption scheme consists of four algorithms: Setup, Ring-Signcrypt, Ring-UnSigncrypt, and Signer Admission algorithms. These algorithms are detailed below.

Setup Algorithm:

The Setup Algorithm generates system public parameters and functions. On the input of an security parameter l, Setup Algorithm outputs two large public primes p and q with $q|(p-1)$, a public generator $g \in Z_p^*$ with order q, and two public hash functions $h_l:\{0,1\}^* \rightarrow \{0,1\}^l$ and $h_q:\{0,1\}^* \rightarrow Z_q^*$. Each ring member has a private key x_i and a public key $y_i = g^{xi} \bmod p$.

Ring-Signcrypt Algorithm

The Ring-Signcrypt Algorithm generates the ring-signcryptext on a message m with a ring $U = \{U_1, U_2, \ldots, U_n\}$ chosen by the real signer $U_s \in U$. After receiving the message m, the ring U, the signer's private key x_s, and the designated verifier U_v's public key y_v, the actual signer U_s performs Ring-Signcrypt algorithm to generate the ring-signcryptext σ, where $U_v \notin U$. This algorithm is described as follows.

Step 1: Choose a random integer $k \in Z_q^*$, compute $K = g^k \bmod p$, $k' = h_q(y_v^k \bmod p)$, and $k_e||k_m = h_l(y_v^k \bmod p||K)$, and encrypt the message m using $C = E_{k_e}(m||h_l(m, k_m))$.

Step 2: Compute all ring member's temporary public keys $y_{k',i} = y_i^{k'} \bmod p$, for $i = 1, 2, \ldots, n$.

Step 3: Forge the other ring member's ElGamal Signature (α_i, β_i) on m_i by first choosing two random numbers $a_i \in Z_q$, $b_i \in Z_q^*$, and computing $\alpha_i = g^{a_i} y_i^{b_i} \bmod p$, $\beta_i = -\alpha_i b_i^{-1} \bmod q$ and $m_i = a_i \beta_i \bmod q$ for $i = 1, 2, \ldots, n$, $i \neq s$.

Step 4: Choose a random integer $r \in Z_q^*$ and compute $v = h_q(g^r \bmod p||m||y_s)$ and $s = r - x_s v \bmod q$. These parameter s should be kept secret.

Step 5: Find m_s by computing

$v_{s+1} = h_q(m, v)$,
$v_{s+2} = h_q(m, v_{s+1} + m_{s+1} \bmod q)$,
$v_{s+3} = h_q(m, v_{s+2} + m_{s+2} \bmod q)$, …, $v_n = h_q(m, v_{n-1} + m_{n-1} \bmod q)$,
$v_1 = h_q(m, v_n + m_n \bmod q)$,
$v_2 = h_q(m, v_1 + m_1 \bmod q)$, …, $v_s = h_q(m, v_{s-1} + m_{s-1} \bmod q)$, and
$m_s = (-v_s) + v \bmod q$.

Step 6: Generate the ElGamal Signature (α_s, β_s) on m_s by choosing a random integer $k_s \in Z_q^*$, and computing $\alpha_s = g^{k_s} \bmod p$ and $\beta_s = k_s^{-1}(m_s - x_s \alpha_s) \bmod q$.

Step 7: Output the ring-signcryptext $\sigma = (C; Y'; K; i_0, v_{i_0}; m_1, m_2, \ldots, m_n; (\alpha_1, \beta_1), (\alpha_2, \beta_2), \ldots, (\alpha_n, \beta_n))$, where i_0 is a random integer in $\{0, 1, 2, \ldots, n\}$ and $Y' = \{y_{k'1}, y_{k'2}, \ldots, y_{k'n}\}$.

Ring-Unsigncrypt Algorithm

After receiving the ring-signcryptext $\sigma = (C; Y'; K; i_0, v_{i_0}; m_1, m_2, \ldots, m_n; (\alpha_1, \beta_1), (\alpha_2, \beta_2), \ldots, (\alpha_n, \beta_n))$, the designated verifier can decrypt and validate σ as follows.

Step 1: Obtain the session key $k_e||k_m = h_l(K^{x_v} \bmod p||K)$ and recover the message and digest by $m||h_l(m, k_m) = D_{k_e}(C)$.

Step 2: Compute the public keys by $k' = h_q(K^{x_v} \bmod p)$ and $y_i = y_{k',i}^{k'^{-1}} \bmod p$ for $i = 1, 2, \ldots, n$.

Step 3: Verify the ElGamal signature (α_i, β_i) on m_i by validating $g^{m_i} = y_i^{\alpha_i} \alpha_i^{\beta_i} \bmod p$ for $i = 1, 2, \ldots, n$.

Step 4: Check $v_{i_0+1} = h_q(m, v_{i_0} + m_{i_0} \bmod q)$,
$v_{i_0+2} = h_q(m, v_{i_0+1} + m_{i_0+1} \bmod q)$,
…,
$v_n = h_q(m, v_{n-1} + m_{n-1} \bmod q)$,
$v_1 = h_q(m, v_n + m_n \bmod q)$,
…, and $v'_{i_0} = h_q(m, v_{i_0-1} + m_{i_0-1} \bmod q)$.
If the equation $v'_{i_0} = v_{i_0}$ holds, then output "accept"; otherwise, reject.

The Ring-Unsigncrypt algorithm outputs "accept" only when the verifications of these five steps are correct.

Admission Algorithm

When the actual signer want to disclose his/her identity for the ring-singcrypt σ, he/she reveals (v, s). The verifier verifies the equation $v = h_q(m||g^s y_s^v||y_s)$. If the equation holds, then the verifier is convinced that the sender of (v, s) is the actual signer.

3. SECURITY ANALYSIS

The proposed scheme is existentially unforgeable against adaptive chosen message attacks (EF-CMA2). In this scheme, the ring-signcryptext is also indistinguishably secure against adaptive chosen ciphertext attacks (IND-CCA2). This scheme also has correctness, strong designated-verifier, signer anonymity, and signer ambiguity properties. The underlying hard problems for the scheme are first described below.

Definition 1: Decision Diffie-Hellman Problem (DDHP)

Let p, q be two large prime numbers with $q|(p-1)$ and g be a generator order q in Z_p^*. Given g^a, g^b, $g^c \pmod p$ to determine whether $g^c \equiv g^{ab} \pmod p$, where a, b, and $c \in Z_q^*$.

DDHP Assumption: No probabilistic polynomial-time (PPT) algorithm solves DDHP with non-negligible probability.

These security properties were proved one by one below.

Correctness (Definition 1): A strong designated verifier ring signcryption scheme is correct if the ring-signcryptext σ generated by Ring-Signcrypt algorithm always passes the verification by Ring-Unsigncrypt algorithm.

The parameters k', the encryption and authenticated keys k_e and k_m, the message, and the public keys can be correctly recovered. Because $y_v^k \equiv K^{x_v} \pmod p$, the verifier recovers the correct $k' = h_q(y_v^k \bmod p)$ and $k_e||k_m = h_l(y_v^k \bmod p||K)$. Since k_e and k_m are correct and $m||h_l(m, k_m) = D_{k_e}(E_{k_e}(m||h_l(m, k_m)))$, the message m and the digest $h_l(m, k_m)$ are recovered correctly. Using the correct k', the public keys $y_i = y_{k',i}^{k'^{-1}} \bmod p$ can be recovered correctly.

Next, the validation of the ElGamal signature (α_i, β_i) on m_i is correct for $i = 1, 2, \ldots, n$. Because $g^{m_i} \equiv g^{a_i\beta_i} \equiv y_i^{\alpha_i} g^{a_i\beta_i} y^{b_i(-\alpha_i b_i^{-1})} \equiv y_i^{\alpha_i}\alpha_i^{\beta_i} \pmod p$, all the ElGamal signatures (α_i, β_i)'s satisfy $g^{m_i} \equiv y_i^{\alpha_i}\alpha_i^{\beta_i} \pmod p$. Finally, the verification for $(i_0, v_{i_0}; m_1, m_2, \ldots, m_n)$ holds because $m_s = (-v_s) + v \bmod q$, and

$$
\begin{aligned}
&v_{i_0+1} = h_q(m, v_{i_0} + m_{i_0} \bmod q),\\
&v_{i_0+2} = h_q(m, v_{i_0+1} + m_{i_0+1} \bmod q),\\
&\ldots, v_n = h_q(m, v_{n-1} + m_{n-1} \bmod q),\\
&v_1 = h_q(m, v_n + m_n \bmod q), \ldots, \text{and}\\
&v_{i_0} = h_q(m, v_{i_0-1} + m_{i_0-1} \bmod q).
\end{aligned}
$$

According to the correctness of k', k_e, k_m, the recovered public keys, ElGamal signatures,

and the received v_{i0}, the Ring-Unsigncrypt algorithm correctly decrypts and validates the ring-signcryptext generated by the Ring-Signcrypt algorithm.

Existential Unforgeability against Adaptive Chosen Message Attacks (EF-CMA2) (Definition 2): In a strong designated verifier ring signcryption scheme, except ring members, no one generates a valid ring-signcryptext for some random message without ring members' private keys, after collecting ring-signcryptexts with chosen messages.

In other words, if some adversary can existentially forge ring-signcryptexts for some messages, the adversary wins the EF-CMA2 game with non-negligible probability. The following discussion describes the EF-CMA2.

EF-CMA2 Game The game consists of two participators: Challenger and Adversary. Challenger performs the Setup algorithm to generate public parameters p, q, and g, and some user's public keys for Adversary. Challenger also provides two hash oracles, h_q and h_l, and one the Ring-Signcrypt oracle. Adversary is then allowed to request those oracles to collect the ring-signcryptext σ for some message chosen by Adversary polynomial times. The h_q oracle, h_l oracle and Ring-Signcrypt oracle are stated sequentially.

***h_q* oracle:** When Adversary queries the h_q oracle, the h_q oracle returns a corresponding hash value to Adversary.

***h_l* oracle:** When Adversary queries the h_l oracle, the h_l oracle returns a corresponding hash value.

Ring-Signcrypt Oracle (RS-Oracle): When Adversary queries Ring-Signcrypt oracle with input (m, y_s, Y), the Ring-Signcrypt oracle returns a ring-signcryptext σ to Adversary, where $Y = \{y_1, y_2, .., y_n\}$ is a set of ring members' public keys.

After collecting many (σ_i, Y, m_i), Adversary outputs a ring-signcryptext σ^* for the message m^* and the ring members' public key set Y, where the message m^* is not queried RS-Oracle before. Adversary wins the game if the ring signcryptext σ^* can pass the Ring-Unsigncrypt.

Theorem 1: A valid ring-signcryptext in the proposed scheme is existentially unforgeable against adaptive chosen message attacks (EF-CMA2) based on the unforgeability of the ElGamal signatures.

***Proof*:** First, Challenger takes a security parameter l for the Setup Algorithm to generate p, q, g, and two hash functions h_q, h_l for Adversary. Adversary can request RS-oracle to generate a ring-signcryptext σ in polynomial times. The oracles h_q, h_l, and RS-oracle are described below.

***h_q* Oracle:** Oracle h_q accepts two types of queries described below.

Case 1: (m, $m_{i-1} + v_{i-1} \bmod q$, $\perp$, $\perp$)-query

Oracle h_q first searches its local records, h_q-list. If it finds the record (m, $m_{i-1} + v_{i-1} \bmod q$, v_i), then h_q returns the digest v_i; otherwise, h_q returns a random value $v_i \in Z_q^*$ and

stores the record (m, $m_{i-1} + v_{i-1} \bmod q$, v_i) in its h_q-list.

Case 2: (m, $m_{s-1} + v_{s-1} \bmod q$, v, m_s)-query

In this case, Oracle h_q computes $v_s = (v + (-m_s)) \bmod q$ and stores the record (m, $m_{s-1} + v_{s-1} \bmod q$, v_s) in its local h_q-list.

h_l Oracle: For any query, h_l oracle first searches its local records, h_l-list. If the query ($y_v^k \bmod p||K$) has been made before, then h_l oracle returns the stored hash value for $y_v^k \bmod p||K$; otherwise, h_l oracle returns a random value $v_l \in \{0, 1\}^l$ and stores the record in its h_l-list.

Ring-Signcrypt Oracle (RS-oracle) On the input (m, y_s, $\{y_1, y_2, ..., y_n\}$), the Ring-Signcrypt oracle outputs a ring-signcryptext σ based on the following procedure, where y_s is the signer's public key and m is a chosen message.

Step 1: Choose a random integer $k \in Z_q^*$, compute $K = g^k \bmod p$, $k' = h_q(y_v^k \bmod p)$, and $k_e||k_m = h_l(y_v^k \bmod p||K)$, and encrypt the message m using $C = E_{k_e}(m||h_l(m, k_m))$.

Step 2: Compute all temporary public keys $y_{k',i} = y_i^{k'} \bmod p$, for $i = 1, 2, \ldots, n$.

Step 3: Forge the other ring member's ElGamal signature (α_i, β_i) on m_i by first randomly choosing $a_i \in Z_q$, $b_i \in Z_q^*$, and computing $\alpha_i = g^{a_i} y_i^{b_i} \bmod p$, $\beta_i = -\alpha_i b_i^{-1} \bmod q$ and $m_i = a_i \beta_i \bmod q$ for $i = 1, 2, \ldots, n$.

Step 4: Choose a random value $v \in Z_q^*$.

Step 5: Find the v_s with the help of h_q oracle answering.

$v_{s+1} = h_q(m, v, \perp, \perp)$,
$v_{s+2} = h_q(m, v_{s+1} + m_{s+1} \bmod q, \perp, \perp)$,
$v_{s+3} = h_q(m, v_{s+2} + m_{s+2} \bmod q, \perp, \perp)$,
…,
$v_n = h_q(m, v_{n-1} + m_{n-1} \bmod q, \perp, \perp)$,
$v_1 = h_q(m, v_n + m_n \bmod q, \perp, \perp)$,
$v_2 = h_q(m, v_1 + m_1 \bmod q, \perp, \perp)$,
…, and $v_s = h_q(m, v_{s-1} + m_{s-1} \bmod q, v, m_s)$,
In this step, the returned value is $v_s = (-m_s) + v \bmod q$.

Step 6: Output the ring-signcryptext $\sigma = (C; Y'; K; i_0, v_{i_0}; m_1, m_2, \ldots, m_n; (\alpha_1, \beta_1), (\alpha_2, \beta_2), \ldots, (\alpha_n, \beta_n))$, where i_0 is a randomly chosen index between 0 and n and $Y' = \{y_{k'1}, y_{k'2}, \ldots, y_{k'n}\}$.
Adversary can query in polynomial times.

If Adversary can generate a valid ring-signcryptext with a non-negligible probability, Challenger can forge an ElGamal signature for a specified message m' without the signer's private key with the help of Adversary. Suppose that the ElGamal signature forgery instance is (m', y). In this case, the challenger generates n public keys $y_i = g^{x_i} \bmod p$ for $i = 1, 2, \ldots, n$ and $i \neq s$ and $y_s = y$.

After collecting many ring-signcryptexts on some messages, Adversary must forge the message m. During the Adversary forgery, the oracle h_q returns the digest $v_s = (-m') + v \bmod q$ for the query (m, $v_{s-1} + m_{s-1} \bmod q$, $\perp$, $\perp$). In other words, Adversary is forced to generate the ElGamal signature on the message $m_s = m'$. Finally, Adversary forges a

valid ring-signcryptext $\sigma = (C; Y'; K; i_0, v_{i_0}; m_1, m_2, \ldots, m_n; (\alpha_1, \beta_1), (\alpha_2, \beta_2), \ldots, (\alpha_n, \beta_n))$ on m, and Challenger obtains the ElGamal signature (α_s, β_s) on the message m'.

Indistinguishable Confidentiality against Adaptive Chosen Ciphertext Attacks (IND-CCA2)

IND-CCA2 game is described before giving the definition IND-CCA2 and Theorem 2 shows that the proposed scheme provides IND-CCA2.

IND-CCA2 Game The IND-CCA2 game has two participators: Challenger G_1 and Adversary A_1. On the security parameter l, Challenger G_1 generates the systems public parameters, p, q, g, and the designated-verifier's public key, and the hash oracle h_l, sign-crypt and unsigncrypt oracles for Adversary A_1. By adaptive chosen ciphertext attacks, Adversary A_1 can collect some ciphertexts with the help of three oracles. These oracles are described below.

***h_l* oracle:** On the query input, the h_l oracle returns a random digest for fresh queries; otherwise, the h_l oracle returns the used digest.

Signcrypt oracle (S-Oracle): On the input (m, y_v, K), the Signcrypt oracle returns a ciphertext C.

Unsigncrypt oracle (U-Oracle): On the input C, the U-Oracle returns a message m and corresponding digest for the ciphertext C.

The game includes two phases: collection and guess phases. In the collection phase, Adversary A_1 collects many of his/her chosen ciphertexts on some messages. In the guessing phase, Adversary A_1 randomly chooses and sends two new messages m_0 and m_1 to Challenger G_1. After randomly choosing one bit $b \in \{0, 1\}$, Challenger G_1 signcrypts m_b, then sends the ciphertext C^* to Adversary A_1. Adversary A_1 may collect more chosen ciphertext after receiving the challenge C^*. Finally, Adversary A_1 gives his/her answer bit b'. Adversary A_1 wins the game if $b' = b$.

Definition 3: A strong designated verifier ring signcryption scheme provides IND-CCA2 if no PPT algorithm wins the IND-CCA2 game with non-negligible advantage over 1/2.

Theorem 2: If Adversary wins the IND-CCA2 game with non-negligible advantage, then a PPT algorithm solving the DDHP with non-negligible probability exists.

***Proof*:** The following discussion describes the three oracles in the IND-CCA2 game.

***h_l* oracle:** The inputs of this oracle are classified into three cases.

Case 1: (y_v^k, K)-query

On the input (y_v, K), h_l oracle first searches its local records, h_{l1}-list. If the query has been made before, h_l oracle returns the found digest $k_e||k_m$ in h_{l1}-list. Otherwise, it returns a new random string $k_e||k_m$ such that the bit length $|(k_e||k_m)| = l$ and stores the record $(y_v^k, K, k_e||k_m)$ in its h_{l1}-list.

Case 2: (m, k_m)-query

On the input (m, k_m), the h_l oracle searches its h_{l2}-list first. If it finds the digest in its h_{l2}-list, it returns the found digest; otherwise, it returns a random digest with bit length l and stores $(m, k_m,$ digest $= h_l(m, k_m))$ in its h_{l2}-list.

Case 3: $(\perp, k_m, m_t)$-query

On the input $(\perp, k_m, m_t)$, the h_l oracle finds a message m and a bit string H, such that $m_t = m||H$. In this case, the oracle stores the record (m, k_m, H) in its h_{l2}-list. Finally, it returns m and the digest H.

Signcrypt Oracle

To respond to the query (m, y_v), the signcrypt oracle searches the S-list. If the query (m, y_v) has already been queried, it returns the corresponding ciphertext. Otherwise, the signcrypt oracle first randomly chooses a value $k \in Z_q^*$, and computes $K = g^k \bmod p$ and $k_e||k_m = h_l(y_v^k \bmod p||K)$. Finally, the signcrypt oracle returns the ciphertext $C = E_{k_e}(m||h_l(m, k_m))$ and records it in the S-list (m, y_v, C).

Unsigncrypt Oracle

On the input (C, y_v), the unsigncrypt oracle searches the S-list first. If the record $(m, C, y_v, k_e||k_m, K)$ in S-list has already been found for the query (C, y_v), the unsigncrypt oracle returns the m and digest $= h_l(m, k_m)$ with the help of the hash oracle h_l. Otherwise, the unsigncrypt oracle chooses a value $k \in Z_q^*$, and computes $K = g^k \bmod p$ and $k_e||k_m = h_l(y_v^k \bmod p||K)$. The unsincrypt oracle then decrypts the cipthertext C by k_e to obtain $m_t = D_{k_e}(C)$. With the help of the hash oracle by the query $(\perp, k_m, m_t)$, the unsigncrypt oracle obtains m and the digest $h_l(m, k_m)$ such that $m_t = m||h_l(m, k_m)$. Finally, the unsigncrypt oracle returns m and the digest, $h_l(m, k_m)$.

If Adversary A_1 wins the game with non-negligible advantage ε, then Challenger G_1 can use A_1 to solve DDHP with non-negligible probability. Suppose that the DDHP instance is $(g^a \bmod p, g^c \bmod p, g^d \bmod p)$. Challenger G_1 sets the public key of the designated verifier $y_v = g^c \bmod p$. In the guessing phase, after receiving the $\{m_0, m_1\}$, Challenger G_1 sets $K = g^a \bmod p$ and computes $(k_e^*||k_m^*) = h_l(g^d \bmod p||K)$. After choosing the random bit b, the ciphertext $C = E_{k_e^*}(m_b)$ and K are sent to A_1. If A_1's guessing is correct, then Challenger G_1 answers that $g^d \bmod p = g^{ac} \bmod p$; otherwise, $g^d \bmod p \neq g^{ac} \bmod p$.

There are two cases in the probability analysis solving the DDHP.

Case 1: $g^d = g^{ac} \bmod p$. Because Adversary A_1 provides the correct answer with probability $(1/2 + \varepsilon)$, Challenger G_1 also solves the yes-instance of DDHP with successful probability $(1/q)(1/2 + \varepsilon)$.

Case 2: $g^d \bmod p \neq g^{ac} \bmod p$. The challenging ciphertext C^* is correct because of the hash collision occurs with probability $(1/2^l)$. Adversary A_1's correct answer causes the incorrect answer for the input no-instance with probability $(1/2 + \varepsilon)$. Conversely, when the hash collision does not occur, the challenging ciphertext is incorrect. For the incorrect ciphertexts, Challenger G_1 determines that the challenging ciphertexts are incorrect when the A_1 is over its polynomial computational time bound. Then Challenger G_1 provides the correct answer that $g^d \bmod p \neq g^{ac} \bmod p$. In this case, Challenger G_1 provides

the correct answer with probability $(1 - (1/q))(1 - (1/2^l)(1/2 + \varepsilon))$.

Finally, Challenger G_1 correctly solves DDHP with probability $(1/q)(1/2 + \varepsilon) + (1 - (1/q))(1 - (1/2^l)(1/2 + \varepsilon)) = [(1/q) + (1 - (1/q))(1 - (1/2^l))](1/2 + \varepsilon) = [1 + (q - 1)(1 - (1/2^l))](1/q)(1/2 + \varepsilon) = [q - (q - 1)(1 - (1/2^l))(1/2 + \varepsilon) > (q - (q - 1))(1/2 + \varepsilon) = (1/2 + \varepsilon)$. Because $(1/2 + \varepsilon)$ is non-negligible, DDHP is solved with non-negligible probability.

Strong Designated-verifier (Definition 4): A strong designated verifier ring signcryption scheme satisfies the strong designated verifier property if only the designated verifier and actual signer can decrypt and verify the ring-signcryptext.

Theorem 3 shows that only the designated verifier and actual signer can decrypt and verify the ring-signcryptext. Message decryption requires the session key k_e obtained by computing $k_e||k_m = h_l(y_v^k \bmod p||K)$. The secret parameter $k' = h_q(y_v^k \bmod p)$ is used to recover temporary public keys. To determine $k_e||k_m$ and k', the common secret item $y_v^k \bmod p = K^{x_v} \bmod p$ is necessary. Therefore, the hardness of DDHP can be used to prove the designated verifier property in Theorem 3.

Theorem 3: Only the designated verifier and actual signer are able to recover the temporary public keys $Y' = \{y_{k',1}, y_{k',2}, \ldots, y_{k',n}\}$ and subsequently decrypt the ring-sincryptext, where $y_{k',i} = y_i^{k'}$ for $i = 1, 2, \ldots, n$. Others cannot recover the public keys and decryption the ring-signcryptext based on the hardness of DDHP.

***Proof*:** The first part of the proof shows that only the designated verifier and the actual signer recover the public keys correctly. Suppose that a PPT Adversary A_2 can recover the actual ring members' public keys with the input (K, y_v, Y'). With the help of Adversary A_2, a PPT Algorithm B is proposed to solve the DDHP problem with non-negligible probability. If the instance of DDHP is $(g^a \bmod p, g^b \bmod p, g^c \bmod p)$, the goal of DDHP is to determine whether $g^c \bmod p = g^{ab} \bmod p$.

Algorithm B first sets that $K \equiv g^a \pmod p$, $y_v \equiv g^b \pmod p$. Then, Algorithm B computes $k^* = h_q(g^c \bmod p)$ and the temporary public keys $y_{k^*,i} = y_i^{k^*} \pmod p$ for $i = 1, 2, \ldots, n$. Next, Algorithm B queries Adversary A_2 the instance $(K, y_v, Y^* = \{y_{k^*,1}, y_{k^*,2}, \ldots, y_{k^*,n}\})$ and Adversary A_2 responses B the public keys $\hat{Y} = (\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_n)$. If $\hat{Y} = Y = \{y_1, y_2, \ldots, y_n\}$, then Algorithm B outputs $g^c \equiv g^{ab} \pmod p$.

The probability analysis that Algorithm B provides the correct answer is shown below. The analysis consists of two cases because of the relationship between $k^* = h_q(g^c \bmod p)$ and $k' = h_q(y_v^k \bmod p) = h_q(g^{ab} \bmod p)$.

Case 1: $g^c \equiv g^{ab} \pmod p$. In this case, $k^* = h_q(g^c \bmod p) = h_q(y_v^k \bmod p) = h_q(g^{ab} \bmod p) = k'$. Because $k^* = k'$, Adversary A_2 correctly responds the public key $\hat{Y} = Y = \{y_1, y_2, \ldots, y_n\}$. Thus, Algorithm B correctly solves the DDHP by determining whether $\hat{Y} = Y = \{y_1, y_2, \ldots, y_n\}$. The correct probability of this case is $(1/q)$.

Case 2: $g^c \bmod p \neq g^{ab} \bmod p$. Because $g^c \bmod p \neq g^{ab} \bmod p$, $k^* = k'$ only because of the hash collision $h_q(y_v^k \bmod p) = h_q(g^c \bmod p)$. With the hash collision $h_q(y_v^k \bmod p) = h_q(g^c \bmod p)$, Algorithm B provides the incorrect answer. Conversely, without hash collision,

Adversary A_2 always provides the incorrect response. Thus, B provides the correct answer, $g^c \bmod p \neq g^{ab} \bmod p$. In this case, Algorithm B provides the incorrect answer with probability $((q-1)/q))(1/q) = (q-1)/q^2$.

Based on this analysis, the probability of an incorrect answer of Algorithm B is $((q-1)/q))(1/q) = (q-1)/q^2 < 1/q$. Because q is a large prime number, the probability $1/q$ is negligible. Hence, Algorithm B solves DDHP with a non-negligible probability.

The second part of this proof shows that no one is able to decrypt the ring-signcryptext except the designated-verifier and actual signer. Suppose that an adversary recover the message from the ring-signcryptext by giving the parameter K and the designated-verifier's public key y_v. By using this adversary, this proof proposes Algorithm B^* to solve the DDHP instance, ($g^a \bmod p$, $g^b \bmod p$, $g^c \bmod p$). Algorithm B^* first sets the $K \equiv g^a \equiv g^k \pmod p$ and $y_v \equiv g^b \equiv g^{x_v} \pmod p$. Algorithm B^* computes $k_e^* || k_m^* = h_l(g^c \bmod p || K)$ and obtains the ring-signcryptext $C^* = E_{k_e^*}(m || h_l(m, k_m^*))$ where m is the message chosen by B. On the input (C^*, K, y_v), the adversary outputs a message m^*. If $m = m^*$, then B^* answers that $g^c \equiv g^{ab} \pmod p$; otherwise, $g^c \bmod p \neq g^{ab} \bmod p$.

Similarly, Algorithm B^* provides the incorrect answer when $g^c \bmod p \neq g^{ab} \bmod p$ and the occurrence of $h_l(g^c \bmod p || K) = h_l(g^{ab} \bmod p || K)$. Thus, the probability of Algorithm B^*'s incorrect answer is $((q-1)/q))(1/2^l) < (1/2^l)$. The probability-bound $(1/2^l)$ is negligible because the hash function h_l avoids collisions with non-negligible probability. Therefore, Algorithm B^* solves DDHP with non-negligible probability.

Signer Ambiguity (Definition 5): A strong designated verifier ring signcryption scheme satisfies the signer ambiguity property if no PPT designated-verifier can identify the actual signer with non-negligible probability.

Suppose that the ring-signcryptext is $\sigma = (C; Y'; K; i_0, v_{i_0}; m_1, m_2, \ldots, m_n; (\alpha_1, \beta_1), (\alpha_2, \beta_2), \ldots, (\alpha_n, \beta_n))$ on the message m. Consider whether the ring-signcryptext $C = E_{k_e}(m || h_l(m, k_m))$ is used to identify the actual signer. The proposed scheme computes the encryption key k_e and authentication key k_m using $k_e || k_m = h_l(y_v^k \bmod p || K)$. Suppose that h_l is an ideal hash function. Because k is a random value uniformly distributing over Z_q^*, K is also uniformly distributes over Z_p. Because the ring-signcryptext C is also uniformly distributed, C cannot be used to identify the actual signer.

Consider whether the temporary public key Y' is used to identify the actual signer. Because k is randomly and uniformly chosen in Z_q^*, $k' = h_q(K^{x_v} \bmod p)$ is also uniformly distributed over Z_q^*, each temporary public keys $y_{k',i} = y_i^{k'}$ is also uniformly distributed for $i = 1, 2, \ldots, n$. Thus, Y' cannot be used to identify the actual signers.

Consider the ElGamal signatures (α_i, β_i) on m_i for $i = 1, 2, \ldots, n$. These ElGamal signatures are classified into forged signature (α_i, β_i) on m_i for $i = 1, 2, \ldots, n$ and $i \neq s$ and the actual signer's signature (α_s, β_s) on m_s. First, consider the distribution of each forged signature (α_i, β_i). Because a_i and b_i are randomly and uniformly chosen over Z_q and Z_q^*, respectively, $\alpha_i = g^{a_i} y_i^{b_i} \bmod p$, $\beta_i = -\alpha_i b_i^{-1} \bmod q$ and $m_i = a_i \beta_i \bmod q$ are also uniformly distributed over Z_p^*, Z_q^* and Z_q^*, respectively. Next, consider the distribution of the actual signer's signature (α_s, β_s) on m_s. Since v_s and v are uniformly distributed over Z_q^*, the message $m_s = v + (-v_s)$ is also uniformly distributed. The random number k_s is also uniformly distributed over Z_q^*; thus, α_s and β_s are also uniformly distributed over Z_p^* and Z_q^*, respectively. Due to the analysis, all the forged signatures (α_i, β_i)'s and actual signer's signature (α_s, β_s) have the same distribution; thus, they are indistinguishable

from each other. So the ElGamal signatures cannot be used to identify the actual signer.

Next, consider v and v_i for $i = 1, 2, \ldots, n$. Suppose that the hash function h_q is an ideal hash function, so $v = h_q(g^r \bmod p \| m \| y_s)$ is uniformly distributed over Z_q^*, where r is an random integer chosen over Z_q^*. Because the hash function is ideal, v_i is also uniformly distributed over Z_q^*.

According to this analysis, no parts of σ can be used to identify the actual signer; therefore, the designated verifier cannot identify the actual signer.

Signer Anonymity (Definition 6): A strong designated verifier ring signcryption scheme satisfies the signer anonymity property if, except for the actual signer and designated verifier, no PPT algorithm finds out the actual signer among all possible one with non-negligible probability.

To provide signer anonymity, the key point is that no PPT one is able to distinguish the legal ring-signcryptext from illegal ring-signcryptexts, which cannot pass the Ring-Unsigncrypt algorithm, with non-negligible probability. Since no one can distinguish the legal/illegal ring-signcryptexts, no one can find out who the actual signer is. The actual signer and the forgers are the possible candidates, so the identity privacy of the actual signer is protected in signer anonymous way.

The following algorithm may generate an illegal ring-signcryptexts.

Illegal-Ring-Signcrypt Algorithm (IRS)

Step 1: Choose a random value $x'_s \in Z_q^*$ and compute $y'_s = g^{x'_s} \bmod p$ such that y'_s is not a certificated public key.

Step 2: Choose another random integer $k \in Z_q^*$, compute $K = g^k \bmod p$, $k' = h_q(y_v^k \bmod p)$, and $k_e \| k_m = h_l(y_v^k \bmod p \| K)$, and signcrypt the message m using $C = E_{k_e}(m \| h_l(m, k_m))$.

Step 3: Compute all ring member's temporary public keys $y_{k',i} = y_i^{k'} \bmod p$, for $i = 1, 2, \ldots, n$, $i \neq s$, and $y_{k',s} = y'^{k'}_s \bmod p$.

Step 4: Forge the other ring member's ElGamal signature (α_i, β_i) on m_i by first choosing two random numbers $a_i \in Z_q$, $b_i \in Z_q^*$, and computing $\alpha_i = g^{a_i} y_i^{b_i} \bmod p$, $\beta_i = -\alpha_i b_i^{-1} \bmod q$, and $m_i = a_i \beta_i \bmod q$ for $i = 1, 2, \ldots, n$, $i \neq s$.

Step 5: Choose a random integer $v \in Z_q^*$.

Step 6: Find m_s by computing

$v_{s+1} = h_q(m, v)$,
$v_{s+2} = h_q(m, v_{s+1} + m_{s+1} \bmod q)$,
$v_{s+3} = h_q(m, v_{s+2} + m_{s+2} \bmod q)$,
$\ldots, v_n = h_q(m, v_{n-1} + m_{n-1} \bmod q)$,
$v_1 = h_q(m, v_n + m_n \bmod q)$,
$v_2 = h_q(m, v_1 + m_1 \bmod q)$,
$\ldots, v_s = h_q(m, v_{s-1} + m_{s-1} \bmod q)$, and
$m_s = (-v_s) + v \bmod q$.

Step 7: Generate the ElGamal signature (α_s, β_s) on m_s by choosing a random integer $k_s \in Z_q^*$, and then computing $\alpha_s = g^{k_s} \bmod p$ and $\beta_s = k_s^{-1}(m_s - x_s \alpha_s) \bmod q$.

Step 8: Output the ring-signcryptext $\sigma = (C; Y'; K; i_0, v_{i_0}; m_1, m_2, \ldots, m_n; (\alpha_1, \beta_1), (\alpha_2, \beta_2), \ldots, (\alpha_n, \beta_n))$, where i_0 is a random integer between 1 and n and $Y' = \{y_{k'1}, y_{k'2}, \ldots, y_{k's} \ldots, y_{k'n}\}$.

To prove signer anonymity, the following subsections define the signer anonymity game and provide proof of the signer anonymity.

Signer Anonymity Game

First, Adversary A_3 gives Challenger G_2 a message m. G_2 chooses a random bit b. If $b = 0$, Challenger G_2 generates the legal ring-signcryptext on the message m using the Ring-Signcrypt algorithm; otherwise, Challenger G_2 generates the illegal ring-signcryptext using the Illegal Ring-Signcypt algorithm or generating in a random manner. After receiving the legal or illegal ring signcryptext from G_2, Adversary A_3 outputs its guessing bit b' in polynomial time. If Adversary A_3 provides the correct answer $b' = b$ with non-negligible advantage ε more than 1/2, then Adversary A_3 wins the game.

Theorem 4: No PPT adversary wins the Signer Anonymity game with non-negligible advantage based on the hardness of DDHP.

***Proof*:** Suppose that the PPT Adversary A_3 is able to win the Signer Anonymity game with non-negligible advantage ε more than 1/2. In this case, Adversary A_3 can construct a polynomial-time algorithm DDHP-A solving the DDHP with non-negligible probability. Assume that the instance of DDHP is ($g^a \bmod p$, $g^c \bmod p$, $g^d \bmod p$).

Algorithm DDHP-A first sets $y_v = g^c \bmod p$. Adversary A_3 is then asked to randomly choose the message m and send m to DDHP-A. After receiving the message m from Adversary A_3, DDHP-A chooses a random bit b. If $b = 0$, DDHP-A uses the Ring-Signcrypt algorithm with $K = g^a \bmod p$ to generate the legal ring-signcryptext. If $b = 1$, DDHP-A generates the illegal ring-signcryptext using the illegal Ring-Signcryptext algorithm or in a random manner. After receiving the ring-signcryptext from DDHP-A, Adversary A_3 responds with its answer bit b'. If the answer bit is correct, DDHP-A outputs $g^d \equiv g^{ac} \pmod p$; otherwise, DDHP-A outputs $g^d \bmod p \neq g^{ac} \bmod p$.

The probability analysis of DDHP-A consists of the following three cases: $g^d \equiv g^{ac} \pmod p$; $g^d \bmod p \neq g^{ac} \bmod p$ with hash collision; and $g^d \bmod p \neq g^{ac} \bmod p$ without hash collision.

Case 1: $g^d \equiv g^{ac} \pmod p$.

In this case, if Adversary A_3 responds with the correct bit, then DDHP-A solves the DDHP correctly. Because Adversary A_3 wins the game with probability $1/2 + \varepsilon$ and the probability of $g^d \equiv g^{ac} \pmod p$ is $1/q$, the probability solving DDHP in this case is $(1/q)(1/2 + \varepsilon)$.

Case 2: $g^d \bmod p \neq g^{ac} \bmod p$ with hash collision.

In Case 2, the hash collision occurs. The probability of $g^d \bmod p \neq g^{ac} \bmod p$ with hash collision is $(1 - (1/q))(1/q)$ based on the ideal hash function assumption. Because the collision occurs, the challenging ring signcryptext is correct either by the Ring-Signcryptext or illegal Ring-Signcryptext algorithms, although $g^d \bmod p \neq g^{ac} \bmod p$. If Adversary A_3 answers incorrectly with probability $(1/2 - \varepsilon)$, then DDHP-A provides the correct answer for the DDHP problem. Thus, the probability of solving DDHP correctly is $(1 - (1/q))(1/q)(1/2 - \varepsilon)$ in Case 2.

Case 3: $g^d \bmod p \neq g^{ac} \bmod p$ without hash collision.

In this case, since no hash collision occurs and $g^d \bmod p \neq g^{ac} \bmod p$, the ring-signcryptext is incorrect or random. Based on the ideal hash function assumption, the probability of $g^d \bmod p \neq g^{ac} \bmod p$ without hash collision is $(1 - (1/q))^2$. Because challenging ring-signcryptext is always incorrect or random, Adversary A_3 answers correctly with probability 1/2. The probability of correctly solving DDHP in this case is $(1 - (1/q))^2 (1/2 + \varepsilon)(1/2)$.

According to the three analyzed cases, the probability of solving DDHP with $(1/q)(1/2 + \varepsilon) + (1 - (1/q))(1/q)(1/2 - \varepsilon) + (1 - (1/q))^2(1/2 + \varepsilon)(1/2) = (1 - (1/q))^2(1/4 + \varepsilon/2) + (1/q) - (1/q)^2(1/2 - \varepsilon) > (1 - (1/q))^2(1/4 + \varepsilon/2)$, where q is a large prime, $1/2 > (1/2 - \varepsilon) > 0$, and $(1/q) - (1/2)(1/q)^2 > 0$. Because q is a large prime and ε is non-negligible, $(1 - (1/q))^2(1/4 + \varepsilon/2)$ is non-negligible. Thus, DDHP-A solves DDHP with non-negligible probability. However, no PPT algorithm solves DDHP with non-negligible probability; therefore, Adversary A_3 cannot exist.

4. COMPARISONS AND DISCUSSIONS

The comparison of security properties of the proposed scheme with those of Han *et al.*, Huang *et al.*, and Selvi *et al.* First, the signer anonymity is considered. Two types of signer anonymity are considered for these ring signcryption schemes. One is the signer anonymity for the designated verifier and one is the signer anonymity for the other. All four schemes provide 1-out-of-n signer anonymity for the designated verifier. However, the proposed scheme provides 1-out-of-infinite signer anonymity for the others, whereas the three other schemes provide only at most 1-out-of-$(n + 1)$ signer anonymity. Therefore, the signer anonymity of the proposed scheme is stronger than the other three schemes.

Because the signer is anonymous, the signer admission is necessary to show who the actual signer is. Among these four schemes, only the proposed scheme provides sig ner admission, so the proposed scheme provides strong benefit protection for actual signers.

For the confidentiality property, the proposed scheme and those of Huang *et al.* and Selvi *et al.* provide indistinguishable confidentiality against adaptive chosen ciphertext attacks (IND-CCA2). However, only the Harn *et al.* scheme provides indistinguishable confidentiality against chosen plaintext attacks (IND-CPA).

Next, consider the message length. Because the proposed scheme uses symmetric encryption to encrypt messages, it does not have a message length restriction. The other three schemes adopt the public key cryptosystems; thus, they suffer the length restriction on messages. Therefore, the proposed scheme removes the message length restriction. Moreover, the encryption/decryption performance in this scheme is better than that in the other three schemes.

Table 1 provides a comparison of the four mentioned schemes. The proposed scheme provides stronger signer anonymity and signer admission to protect the actual signer's privacy. Conversely, this scheme satisfies IND-CCA2 and EF-CMA2 to provide ring-signcrytexts simultaneously.

Table 2 shows the performance of the proposed scheme. The notations are defined first. Notation T_e is the computation cost to perform one modular exponentiation opera-

tion. T_{inv} is the computation cost to perform one modular inverse operation. T_h is the computation cost to execute hash function once and T_{sym} is the computation cost to perform the symmetric encryption/decryption operation.

In the Ring-Signcryption algorithm, the computation cost of Step 1 is $2T_e + 2T_h + 1T_{sym}$. The computation cost of Step 2 is $n \times T_e$ and the cost of Step 3 is $(n - 1) \times (1.16T_e + 1T_{inv})$ [21]. The computation cost of Step 4 is $1T_e + 1T_h$. The computation cost of Step 5 is $n \times T_h$. The computation cost of Step 6 is $1T_e + 1T_{inv}$. Finally, the Ring-Signcryption algorithm's computation cost is $(2.16n + 2.84)T_e + nT_{inv} + (n + 3)T_h + 1T_{sym}$.

In the Ring-Unsigncrypt algorithm, the computation cost of Step 1 is $1T_{sym} + 1T_e + 1T_h$. The computation cost of Step 2 is $1T_h + 1T_{inv} + nT_e$. In Step 3, the computation cost is $n \times (2.16T_e)$ [21]. In Step 4, the computation cost is nT_h. Finally, the total cost of Ring-Unsigncrypt algorithm is $(3.16n + 1)T_e + 1T_{inv} + (n + 2)T_h + 1T_{sym}$.

The computation cost of admission algorithm is at least $2T_e + 1T_h$. The ring signcryptext must be validated by the Ring-Unsigncrypt algorithm in advance. The extra computation cost is to validate the Schnorr signatures.

Table 1. Security property comparison.

Schemes / Properties	Han *et al.* Scheme	Huang *et al.* Scheme	Selvi *et al.* Scheme	Our Scheme
Signer Anonymity for the Others	$1/(n + 1)$	$1/n$	$1/n$	1/max
Signer Anonymity for the Designated-Verifier	$1/n$	$1/n$	$1/n$	$1/n$
Signer Admission	No	No	No	Yes
Message Confidentiality	IND-CPA	IND-CCA2	IND-CCA2	IND-CCA2

Table 2. Computation cost of our scheme.

Algorithms	Our Scheme
Ring-Signcrypt	$(2.16n + 2.84)T_e + nT_{inv} + (n + 3)T_h + 1T_{sym}$
Ring-Unsigncrypt	$(3.16n + 1)T_e + 1T_{inv} + (n + 2)T_h + 1T_{sym}$
Admission	$\geq 2T_e + 1T_h$

5. CONCLUSIONS

A ring signature scheme with a strong designated verifier provides signer anonymity to protect the signer's identity. However, the message may reveal some sensitive data about the signer's identity. To remove this flaw, Han *et al.*, Huang *et al.*, and Selvi *et al.* each proposed ring signcryption schemes. To provide stronger signer anonymity, this study proposes a ring signcryption scheme with a (strong) designated verifier. Compared with the Han *et al.*, Huang *et al.*, and Selvi *et al.* schemes, this scheme provides stronger signer anonymity for the other verifiers. This scheme still provides the same signer anonymity for the designated verifier as the other three schemes. For unforgeability and confidentiality properties, this scheme is as strong as the Huang *et al.* scheme and Selvi *et al.* scheme, but stronger than the Han *et al.* scheme. The proposed scheme is also free of the message length restriction, whereas the other three schemes are not.

REFERENCES

1. R. L. Rivest, A. Shamir, and Y. Tauman, "How to leak a secret," *Advances in Cryptology-ASIACRYPT*, LNCS 2248, 2001, pp. 552-565.
2. F. Zhang and K. Kim, "ID-based blind signature and ring signature form pairings," *Advance in Cryptology-Asiacrypt*, LNCS 2501, 2002, pp. 629-637.
3. A. Shamir, "Identity-based cryptosystems and signature schemes," *Advance in Croptology-Crypto*, 1985, pp. 47-53.
4. 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, 2004, pp. 325-335.
5. E. Bresson, J. Stern, and M. Szydlo, "Threshold ring signatures and applications to Fdad-hoc groups," *Advances in Cryptology-Crypto*, LNCS 2442, 2002, pp. 465-480.
6. J. Ren and L. Harn, "Generalized ring signature," *IEEE Transactions on Dependable and Secure Computing*, Vol. 5, 2008, pp. 155-163.
7. H. Wang, S. Han, C. Deng, and F. Zhang, "Cryptanalysis and improvement of a ring signature based on ElGamal signature," in *Proceedings of WRI World Congress on Software Engineering*, Vol. 3, 2009, pp. 397-401.
8. J. S. Lee and J. H. Chang, "Strong designated verifier ring signature scheme," *Innovations and Advanced Techniques in Computer and Information Sciences and Engineering*, Springer-Verlag, NY, 2007, pp. 543-547.
9. L. Wu and D. Li, "Strong designated verifier ID-based ring signature scheme," in *Proceedings of International Symposium on Information Science and Engineering*, Vol. 1, 2008, pp. 294-298.
10. S. J. Hwang and J. Y. Chen, "An efficient strong designated-verifier ring signature scheme providing maximal signer anonymity," in *Proceedings of National Computer Symposium on Cryptography and Information Security*, 2011, pp. 234-244.
11. Y. Zheng, "Digital signcryption or how to achieve cost (signature and encryption) cost (signature) + cost (encryption)," *Advances in Crypto'97*, LNCS 1294, 1997, pp. 165-179.
12. X. Huang, W. Susilo, Y. Mu, and F. Zhang, "Identity-based ring signcryption schemes: Cryptographic primitives for preserving privacy and authenticity in the ubiquitous world," in *Proceedings of Advanced Information Networking and Applications*, 2005, pp. 649-654.
13. S. Han, H. Wang, and X. Wang, "A strong designated verifier ring signcryption scheme," in *Proceedings of the 5th International Conference on Wireless Communications, Networking and Mobile Computing*, 2009, pp. 4450-4453.
14. S. Sharmila D. Selvi, S. S. Vivek, and C. P. Rangan, "On the security of identity based ring signcryption scheme," *Information Security*, LNCS 5735, 2009, pp. 310-325.
15. Y. Yu, F. Li, C. Xu, and Y. Sun, "An efficient identity-based anonymous signcryption scheme," *Wuhan University Journal of Natural Sciences*, Vol. 13, 2008, pp. 670-674.
16. Z.-C. Zhu, Y.-Z. Zhang, and F. Wang, "An efficient and provable secure identity based ring signcryption scheme," *Computer Standards and Interfaces*, Vol. 31, 2009, pp. 1092-1097.

17. M. Zhang, B. Yang, S. Zhu, and W. Zhang, "Efficient secret authenticatable anonymous signcryption scheme with identity privacy," in *Proceedings of IEEE International Workshops on Intelligence and Security Informatics*, 2008, pp. 126-137.
18. F. Li, H. Xiong, and Y. Yu, "An efficient Id-based ring signcryption scheme," *Communications, Circuits and Systems*, 2008, pp. 483-487.
19. F. Li, M. Shirase, and T. Takagi, "Analysis and improvement of authenticatable ring signcryption scheme," *Journal of Shanghai Jiaotong University*, Vol. 13, 2008, pp. 679-683.
20. L. Zhun and F. Zhang, "Efficient identity based ring sgnature and ring signcryption schemes," in *Proceedings of International Conference on Computational Intelligence and Security*, 2008, Vol. 2, pp. 303-307.

Shin-Jia Hwang (黃心嘉) is currently an Asociate Pofessor of Department of Computer Science and Information Engineering, Tamkang University, Tamsui, New Taipei City, Taiwan. During the academic years of 1996-2001, he was on the faculty of the Department of Information Management at Chaoyang University of Technology, WuFeng, New Taichung City, Taiwan. He received his B.S. degree in Information and Computer Engineering from Chung-Yuan Christian University, Chungli, Taiwan in 1987 and his MS degree in Computer Science and Information Engineering from National Chung Cheng University, Chiayi, Taiwan in 1992. He received his Ph.D. degree in Computer and Information Science form National Chiao Tung University, Hsinchu, Taiwan. His research interests include applied cryptography, computer security and related issues.

Jyong-Ye Chen (陳炯燁) received his BS and MS degrees in Computer Science and Information Engineering from Tamkang University, Tamsui, New Taipei City, Taiwan, in 2007 and 2010 respectively. He is currently working in American Megatrends. His main research interests include apply cryptography and network security.