Efficient and Provably Secure Multi-Recipient Signcryption from Bilinear Pairings

(October 7, 2006)

Fagen Li[1], Yupu Hu[1], and Shuanggen Liu[1,2]

1. Key Laboratory of Computer Networks and Information Security, Xidian University, Xi'an 710071, Shaanxi, China;

2. College of Computer Information Engineering, Jiangxi Normal University, Nanchang 330022, Jiangxi, China

E-mail: fagenli@mail.xidian.edu.cn

Abstract: Signcryption is a cryptographic primitive that performs signature and encryption simultaneously, at a lower computational costs and communication overheads than the signature-then-encryption approach. In this paper, we propose an efficient multi-recipient signcryption scheme based on the bilinear pairings which broadcasts a message to multiple users in a secure and authenticated manner. We prove its semantic security and unforgeability under the Gap Diffie-Hellman problem assumption in the random oracle model. The proposed scheme is more efficient than re-signcrypting a message n times using a signcryption scheme in terms of computational costs and communication overheads.

Keywords: signcryption; multi-recipient signcryption; bilinear pairings; provable security

1 Introduction

Confidentiality, integrity, non-repudiation and authentication are the important requirements for many cryptographic applications. A traditional approach to achieve these requirements is to sign-then-encrypt the message. Signcryption, first proposed by Zheng[1] in 1997, is a cryptographic primitive that performs signature and encryption simultaneously, at a lower computational costs and communication overheads than the signature-then-encryption approach. Several efficient signcryption schemes have been proposed since 1997. The original scheme in [1] is based on the discrete logarithm problem but no security proof is given. Zheng's original construction[1] was only proven secure in 2002 by Baek et al.[2] who described a formal security model in a multi-user setting.

The bilinear pairings, namely the Weil pairing and the Tate pairing of algebraic curves, are im-

[*] Full version of a paper that will be published in the 2nd Chinese Conference on Trusted Computing and Information Security (CTCIS'06), Baoding, China, October, 2006.

portant tools for construction of identity-based (ID-based) cryptographic schemes. Many ID-based signcryption schemes[3−15] and certificate-based signcryption schemes[16−19] using the bilinear pairings have been proposed. However, all of the above schemes consist of only single recipient. In practice, broadcasting a message to multiple users in a secure and authenticated manner is an important facility for a group of people who are jointly working on the same project to communicate with one another. In [20], Zheng proposed a signcryption scheme for multiple recipients (multi-recipient signcryption). The basic idea is to use two types of keys: the first type consists of only a single randomly chosen key (a message-encryption key) and the second type of keys include a key chosen independently at random for each recipient (called a recipient specific key). The message-encryption key is used to encrypt a message with a private key cipher, while a recipient specific key is used to encrypt the message-encryption key. In [21], Seo and Kim proposed a domain-verifiable signcryption scheme which signcrypts n messages to n users. Each user with domain can decrypt just his own message and all users can verify the whole transaction. However, both [20] and [21] are inefficient since they are sequential composition of signcryption. In addition, the formal model and security proof for their schemes are also not considered.

The main contribution of this paper is to present the formal security model for multi-recipient signcryption and propose an efficient multi-recipient signcryption scheme using the bilinear pairings. We then prove its security in the random oracle model assuming the Gap Diffie-Hellman problem is computationally hard.

The rest of this paper is organized as follows. Some preliminary works are given in Section 2. The formal model for multi-recipient signcryption is given in Section 3. The proposed multi-recipient signcryption scheme is described in Section 4. We analyze the proposed scheme in Section 5. Finally, the conclusions are given in Section 6.

2 Preliminaries

In this section, we briefly describe the basic definition and properties of the bilinear pairings.

Let G_1 be a cyclic additive group generated by P, whose order is a prime q, and G_2 be a cyclic multiplicative group of the same order q. Let a, b be elements of ${Z_q}^*$. A bilinear pairings is a map $\hat{e} : G_1 \times G_1 \rightarrow G_2$ with the following properties:

1. Bilinearity: $\hat{e}(aP, bQ) = \hat{e}(P, Q)^{ab}$.

2. Non-degeneracy: There exists P and $Q \in G_1$ such that $\hat{e}(P, Q) \neq 1$.

3. Computability: There is an efficient algorithm to compute $\hat{e}(P,Q)$ for all P,$Q \in G_1$.

The modified Weil pairing and the Tate pairing[22] are admissible maps of this kind. The security of our scheme described here relies on the hardness of the following problems.

Definition 1. Given two groups G_1 and G_2 of the same prime order q, a bilinear map $\hat{e} : G_1 \times G_1 \rightarrow G_2$ and a generator P of G_1,

- The Computational Diffie-Hellman problem (CDH) in G_1 is, given (P, aP, bP) for unknown $a, b \in Z_q$, to compute $abP \in G_1$.

- The Decisional Diffie-Hellman problem (DDH) is, given (P, aP, bP, cP) for unknown $a, b, c \in Z_q$, to decide whether $ab \equiv c(\text{mod}q)$ or not. Tuples of the form (P, aP, bP, cP) for which the latter condition holds are called "Diffie-Hellman tuples".

- The Gap Diffie-Hellman problem (GDH) is to solve a given instance (P, aP, bP) of the CDH problem with the help of a DDH oracle that is able to decide whether a tuple $(P, a'P, b'P, c'P)$ is such that $c' \equiv a'b'(\text{mod}q)$.

As shown in [23], a pairing can implement a DDH oracle. Indeed, in a group G_1 for which pairings are efficiently computable, to determine whether a tuple (P, aP, bP, cP) is a valid Diffie-Hellman tuple or not, it suffices to check if $\hat{e}(P, cP) = \hat{e}(aP, bP)$. This kind of group, where the DDH problem is easy while the CDH one is still believed to be hard, is called Gap Diffie-Hellman groups.

3 Formal Model for Multi-Recipient Signcryption

3.1 Generic Scheme

A generic multi-recipient signcryption scheme for broadcasting a single message consists of the following three algorithms.

Keygen: Given a security parameter k, it generates a private/public key pair (sk, pk).

Signcrypt: Given a message m, a private key sk_S and multiple public keys $pk_{R_1}, \ldots, pk_{R_n}$, it outputs a ciphertext σ. m is drawn from a message space M which is defined as $\{0, 1\}^{n_1}$.

Unsigncrypt: Given a ciphertext σ, a private key $sk_{R_i}(i \in \{1, \ldots, n\})$ and a public key pk_S, it outputs the original message m or the symbol $\perp$ if σ is not a valid ciphertext corresponding to (sk_{R_i}, pk_S).

For consistency purposes, we of course require that if $\sigma = \textbf{Signcrypt}(m, sk_S, pk_{R_1}, \ldots, pk_{R_n})$, then we have $m = \textbf{Unsigncrypt}(\sigma, pk_S, sk_{R_i})$.

3.2 Security Notions

Baek et al.[2] defines the security notions for signcryption schemes. These notions are indistinguishability against adaptive chosen ciphertext attacks and unforgeability against adaptive chosen messages attacks. We modify this definition slightly to adapt for our multi-recipient signcryption scheme.

Definition 2 (Confidentiality). A multi-recipient signcryption scheme is semantically secure against adaptive chosen ciphertext attack (MRSC-IND-CCA2) if no polynomially bounded adversary has a non-negligible advantage in the following game:

1. The challenger runs **Keygen** to generate multiple key pairs $(sk_{R_i}, pk_{R_i})(i = 1, \ldots, n)$. sk_{R_i} is kept secret while pk_{R_i} is given to adversary A.

2. In the first stage, A makes a number of queries to the following oracles:

   - Signcryption oracle: A produces a message $m \in M$ and requires the result of the operation $\textbf{Signcrypt}(m, sk_S, pk_{R_1}, \ldots, pk_{R_n})$.
   - Unsigncryption oracle: A produces a ciphertext σ and an arbitrary public key pk_U, and requires the result of the operation $\textbf{Unsigncryt}(\sigma, pk_U, sk_{R_i})$.

   These queries can be asked adaptively: each query may depend on the answers to previous ones.

3. A produces two plaintexts $m_0, m_1 \in M$. The challenger picks a bit $b \in_R \{0, 1\}$ and computes a signcryption $\sigma^* = \textbf{Signcrypt}(m_b, sk_S, pk_{R_1}, \ldots, pk_{R_n})$ of m_b with the sender's private key sk_S under the attacked receivers' public keys $pk_{R_i}(i = 1, \ldots, n)$. σ^* is sent to A as a challenge ciphertext.

4. A makes a number of new queries as in the first stage with the restriction that it cannot query the unsigncryption oracle with σ^*.

5. At the end of the game, A outputs a bit b' and wins if $b' = b$.

A's advantage is defined to be $Adv^{ind-cca2}(A) := 2\Pr[b' = b] - 1$.

Definition 3 (Unforgeability). A multi-recipient signcryption scheme is existentially unforgeable against chosen message attack (MRSC-EUF-CMA) if no polynomially bounded adversary has a non-negligible advantage in the following game:

1. The challenger runs **Keygen** to generate a key pair (sk_S, pk_S). sk_S is kept secret while pk_S is given to forger F.

2. The forger F makes a number of queries to the signcryption oracle as in the confidentiality game. Again, these queries can also be produced adaptively. Note that we allow F to have access to all recipients' private keys as well as the corresponding public keys.

3. At the end of the game, F produces a ciphertext σ and wins the game if the result of **Unsigncrypt**$(\sigma, pk_S, sk_{R_i})(i = 1, \ldots, n)$ is not the $\perp$ symbol such that σ was not the output of a signcryption query **Signcrypt**$(m, sk_S, pk_{R_1}, \ldots, pk_{R_n})$ made during the game.

Note that we do not require the unsigncryption query since the adversary can simulate the unsigncryption oracle by himself.

4 An Efficient Multi-Recipient Signcryption Scheme

In this section, we propose a certificate-based multi-recipient signcryption scheme using the bilinear pairings. Our scheme is motivated by Yang et al.'s signcryption schem[17].

We assume that both the sender and the recipients agree on public parameters: security parameters k and l, cyclic groups G_1 and G_2 of prime order $q \geq 2^k$ such that l is the number of bits required to represent elements of G_1 , a generator P of G_1 and a bilinear map $\hat{e} : G_1 \times G_1 \to G_2$. They also agree on three hash functions $H_1 : G_1 \to \{0,1\}^{n_1}$, $H_2 : \{0,1\}^{n_1+(n+1)l} \to G_1$ and $H_3 : {G_1}^3 \to \{0,1\}^l$. The proposed scheme consists of the following three algorithms (we recall that the symbol $\oplus$ denotes the bitwise exclusive OR).

Keygen: User u chooses his private key x_u from Z_q randomly and sets corresponding public key $Y_u = x_u P$. We will denote the sender and the recipients respectively by $u = S$ and $u = R_i (i = 1, \ldots, n)$ and their key pair by (x_S, Y_S) and (x_{R_i}, Y_{R_i}).

Signcrypt: To signcrypt a message $m \in M$ for recipients $R_1, \ldots, R_n$, the sender S follows the steps below.

1. Choose $r \in Z_q$ and $R \in G_1$ randomly, respectively.

2. Compute $U = rP$.
3. Compute $c = m \oplus H_1(R)$.
4. Compute $V = x_S H_2(c, U, Y_{R_1}, \ldots, Y_{R_n})$.
5. Compute $Z_i = R \oplus H_3(U, Y_{R_i}, rY_{R_i})$ for $i = 1, \ldots, n$.

The ciphertext is $\sigma = (U, c, V, Z_1, \ldots, Z_n)$.

Unsigncrypt: When receiving a ciphertext $\sigma = (U, c, V, Z_1, \ldots, Z_n)$, the receiver R_i follows the steps below.

1. Compute $R = Z_i \oplus H_3(U, Y_{R_i}, x_{R_i} U)$.
2. Compute $m = c \oplus H_1(R)$.
3. Compute $H = H_2(c, U, Y_{R_1}, \ldots, Y_{R_n})$
4. Accept the message if and only if $\hat{e}(P, V) = \hat{e}(Y_S, H)$, return $\perp$ otherwise.

The consistency of the scheme is easy to verify. Any third party can be convinced of the message's origin by computing $H = H_2(c, U, Y_{R_1}, \ldots, Y_{R_n})$ and checking if the condition $\hat{e}(P, V) = \hat{e}(Y_S, H)$ holds. The knowledge of the plaintext m is not required for the public verification of a message's origin. Therefore, our scheme provides the ciphertext authenticity[5] which is very useful in firewalls[24]. If required, the anonymity property is obtained by scrambling the sender's public key Y_S together with the message at step 3 of **Signcrypt**(i.e. $c = m||Y_S \oplus H_1(R)$.) in such a way that the recipient retrieves it at step 2 of **Unsigncrypt**(i.e. $m||Y_S = c \oplus H_1(R)$).

5 Analysis of the Scheme

In this section, we analyze the security and efficiency of our scheme.

5.1 Security

Theorem 1. In the random oracle model, if an adversary A has a non-negligible advantage ϵ against the MRSC-IND-CCA2 security of the above scheme when running in a time t and performing q_{sc} signcryption queries, q_{usc} unsigncryption queries and q_{H_i} queries to oracles H_i ($i = 1, 2, 3$), then there exists an algorithm B that can solve the CDH problem in the group G_1 with a probability $\epsilon' \geq \epsilon - \frac{q_{H_3} q_{usc}}{2^{2k}}$ in a time $t' \leq t + (2q_{usc} + 2q_{H_3})t_e$, where t_e denotes the time required for one pairing computation.

Proof. The algorithm B runs A as a subroutine to solve the CDH problem in a polynomial time. Let (aP, bP) be a random instance of the CDH problem. B simulates A's challenger in the game of Definition 2 and starts it with $Y_u = bP$ as the challenge public key. Without loss of generality, we let $Y_u = Y_{R_1}$. A then adaptively performs queries as explained in the definition. To handle these queries, B maintains lists L_i to keep track of the answers given to oracle queries on H_i for $i = 1, 2, 3$.

- H_1 queries: For a $H_1(R_e)$ query, B first checks if the value of H_1 was previously defined for the input R_e. If it was, the previously defined value is returned. Otherwise, B randomly chooses g from $\{0,1\}^{n_1}$, returns g as an answer and inserts the tuple (R_e, g) into L_1.

- H_2 queries: For a $H_2(c_e, U_e, Y_{R_1}, \ldots, Y_{R_n})$ query, B first checks if the value of H_2 was previously defined for the input $(c_e, U_e, Y_{R_1}, \ldots, Y_{R_n})$. If it was, the previously defined value is returned. Otherwise, B chooses w from Z_q randomly, returns wP as an answer and inserts the tuple $(c_e, U_e, Y_{R_1}, \ldots, Y_{R_n}, w)$ into L_2.

- H_3 queries: For a $H_3(U_e, Y_{R_i}, P_e)$ query, B first checks if the value of H_3 was previously defined for the input (U_e, Y_{R_i}, P_e). If it was, the previously defined value is returned. Otherwise, B randomly chooses Q from G_1, returns Q as an answer and inserts the tuple (U_e, Y_{R_i}, P_e, Q) into L_3.

- Signcryption queries: For a signcryption query on a plaintext m chosen by the adversary A, B first randomly chooses $r \in Z_q$ and $R \in G_1$, computes $U = rP$, runs the H_1 simulation process to obtain $h_1 = H_1(R)$, computes $c = m \oplus h_1$, and checks if L_2 contains a tuple $(c, U, Y_{R_1}, \ldots, Y_{R_n}, w')$ indicating that $H_2(c, U, Y_{R_1}, \ldots, Y_{R_n})$ was previously defined to be $w'P$. If no such tuple is found, B chooses w' from Z_q randomly and puts the entry $(c, U, Y_{R_1}, \ldots, Y_{R_n}, w')$ into L_2. B then computes $V = w'Y_S = x_S H_2(c, U, Y_{R_1}, \ldots, Y_{R_n})$ for the random w' chosen or recovered from L_2. Finally, B runs the H_3 simulation process to obtain $h_{3_i} = H_3(U, Y_{R_i}, rY_{R_i})$ and computes $Z_i = R \oplus h_{3_i}$ for $i = 1, \ldots, n$. $(U, c, V, Z_1, \ldots, Z_n)$ is then returned as the signcryption of m.

- Unsigncryption queries: For a unsigncryption query on a ciphertext $(U, c, V, Z_1, \ldots, Z_n)$ and a sender's public key Y_S both chosen by A, B proceeds as follows: it scans the list L_3, looking for tuples $(U, Y_{R_1}, S_i, h_{3_i})$ $(0 \le i \le q_{H_3})$ such that $R_i = Z_1 \oplus h_{3_i}$ exists in an entry (R_i, h_{1_i}) of L_1, and for the corresponding elements h_{1_i}, $m_i = c \oplus h_{1_i}$ is such that there exists an entry

$(c, U, Y_{R_1}, \ldots, Y_{R_n}, h_{2_i})$ in the list L_2 satisfying $\hat{e}(P, V) = \hat{e}(Y_S, h_{2_i})$. If no such tuples are found, the $\perp$ symbol is returned to A. Otherwise, m_i is returned to A.

At the end of the first stage, A outputs two plaintexts m_0 and m_1 and requires a challenge ciphertext built under the recipient's public key Y_u. B chooses a random bit $b \in \{0, 1\}$ and signcrypts m_b. To do so, B sets $U^* = aP$ and randomly chooses R^* from G_1, runs the H_1 simulation process to obtain ${h_1}^* = H_1(R^*)$, computes $c_b = m_b \oplus {h_1}^*$, runs the H_2 simulation process to obtain ${h_2}^* = H_2(c_b, U^*, Y_{R_1}, \ldots, Y_{R_n})$, and computes $V^* = {h_2}^* Y_S$. Then B randomly picks ${Z_1}^*, \ldots, {Z_n}^*$ from distributions. Finally, B sends the challenge ciphertext $\sigma = (U^*, c_b, V^*, {Z_1}^*, \ldots, {Z_n}^*) = (aP, c_b, V^*, {Z_1}^* \ldots, {Z_n}^*)$ to A. A performs a second series of queries at a second stage. These queries are handled by B as those at the first stage. It is easy to show that A will not realize that σ is not a valid signcryption for the sender's private key x_S and the public key Y_u unless it asks for the hash value $H_3(aP, bP, abP)$. In that case, the solution of the Diffie-Hellman problem would be inserted in L_3 exactly at that moment and it does not matter if the simulation of A's view is no longer perfect.

At the end of the game, A produces a result which is ignored by B. The latter just looks into the list L_3 for tuples of the form $(aP, bP, D_i, .)$. For each of them, B checks whether $\hat{e}(P, D_i) = \hat{e}(aP, bP)$ and, if this relation holds, stops and outputs D_i as a solution of the CDH problem. If no tuple of this kind satisfies the latter equality, B stops and outputs "failure".

We can now assess B's probability of success. Let AskH_3 be the event that A asks the hash value of abP during the simulation. As long as the simulation of the attack's environment is perfect, the probability for AskH_3 to happen is the same as in a real attack. In a real attack, we have

$$\Pr[b = b'] \leq \Pr[b = b' | \neg\mathrm{AskH}_3]\Pr[\neg\mathrm{AskH}_3] + \Pr[\mathrm{AskH}_3] = \frac{1}{2} + \frac{1}{2}\Pr[\mathrm{AskH}_3]$$

and then we have $\epsilon = 2\Pr[b = b'] - 1 \leq \Pr[\mathrm{AskH}_3]$. Now, the probability that the simulation is not perfect remains to be assessed. The only case where it can happen is when a valid ciphertext is rejected in a unsigncryption query. It is easy to see that for every tuple $(U, Y_{R_1}, S_i, h_{3_i})$ in L_3, there is exactly one pair (h_{1_i}, h_{2_i}) of elements in the range of oracles H_1 and H_2 providing a valid ciphertext. The probability to reject a valid ciphertext is thus not greater than $q_{H_3}/2^{2k}$. So we have

$$\epsilon' \geq \epsilon - \frac{q_{H_3} q_{usc}}{2^{2k}}.$$

The bound on B's computation time derives from the fact that every unsigncryption query requires at most 2 pairing evaluations while the extraction of the solution from L_3 implies to compute at most $2q_{H_3}$ pairings.

Theorem 2. In the random oracle model, if an adversary F that has a non-negligible advantage ϵ against the MRSC-EUF-CMA security of the above scheme when running in a time t and performing q_{sc} signcryption queries, q_{usc} unsigncryption queries and q_{H_i} queries on oracles H_i $(i = 1, 2, 3)$, then there exists an algorithm B that can solve the Diffie-Hellman problem in G_1 with a probability $\epsilon' \geq \epsilon - \frac{q_{sc}q_{H_2}}{2^k}$ in a time $t' = t$.

Proof. B receives a random instance (aP, bP) of the Diffie-Hellman problem. It uses F as a subroutine to solve that instance and plays the role of F's challenger in the game of Definition 3. It initializes F with $Y_u = bP$ as a challenge public key. F then performs adaptive queries that are handled like explained below (using lists L_1, L_2, L_3 as in the proof of Theorem 1):

- H_1 and H_3 queries: H_1 and H_3 queries are dealt with in the usual way as in the proof of Theorem 1.

- H_2 queries: When F asks the hash value of a tuple $(c, U, Y_{R_1}, \ldots, Y_{R_n})$ that was previously queried, B returns the value defined at the previous query. For a query on a new tuple $(c, U, Y_{R_1}, \ldots, Y_{R_n})$, B picks a random $w \in Z_q$ and defines the value of $H_2(c, U, Y_{R_1}, \ldots, Y_{R_n})$ to be $w(aP)$ which is returned to F.

- Signcryption queries: For a signcryption query on a message chosen by F, B picks $r \in Z_q$ and $R \in G_1$ randomly, computes $U = rP$ and $c = m \oplus H_1(R)$ (where the value H_1 is obtained from oracle simulation algorithm). If the value of H_2 is already defined at $(c, U, Y_{R_1}, \ldots, Y_{R_n})$, then B outputs "failure" and halts. Otherwise, B picks a random $w \in Z_q$ and sets $H_2(c, U, R_1, \ldots, Y_{R_n}) = wP$. B then computes $V = wY_S$ and $Z_i = R \oplus H_3(U, Y_{R_i}, rY_{R_i})$ for $i = 1, \ldots, n$ (where the value H_3 is obtained from oracle simulation algorithm). The ciphertext $(U, c, V, Z_1, \ldots, Z_n)$ is then returned to F as the signcryption of m.

At the end of the game, F produces a ciphertext $(U', c', V', Z_1', \ldots, Z_n')$. If the forgery is valid, we have $\hat{e}(P, V') = \hat{e}(Y_u, H_2(c', U', Y_{R_1}, \ldots, Y_{R_n}))$. If the hash value $H_2(c', U', Y_{R_1}, \ldots, Y_{R_n})$ was not asked by F during the simulation, B outputs "failure" and stops. Otherwise, the hash value $H_2(c', U', Y_{R_1}, \ldots, Y_{R_n})$ must have been defined to be $w(aP)$, for some $w \in Z_q$ which is known to B, and V' must be equal to $w(abP)$ that can then easily extract the solution $w^{-1}V'$ of the CDH problem in G_1. We can now assess B's probability of success. It is easy to see that the probability for B to fail in answering a signcryption query is not greater than $q_{sc}q_{H_2}/2^k$ (since at each signcryption query, there is at most q_{H_2} elements in L_2 and the randomly chosen $r \in Z_q$ is uniformly taken from a set of 2^k elements). Therefore, we have

$$\epsilon' \geq \epsilon - \frac{q_{sc} q_{H_2}}{2^k}.$$

5.2 Efficiency

We compare the major computational costs and communication overheads (the length of the ciphertext) of our scheme with those of the obvious construction of multi-recipient signcryption that simply re-signcrypts a message n times using a signcryption scheme. To signcrypt a message m, our scheme only needs $n+2$ scalar multiplications in G_1 (to compute rP, $x_S H_2(c, U, Y_{R_1}, \ldots, Y_{R_n})$, and rY_{R_i}) and $n+2$ hash functions (to compute $H_1(R)$, $H_2(c, U, Y_{R_1}, \ldots, Y_{R_n})$ and $H_3(U, Y_{R_i}, rY_{R_i})$). The ciphertext is $(n+2)|G_1| + |m|$. On the other hand, re-signcrypting a message n times using Yang et al.'s scheme[17] (RSY approach for short) needs $3n$ scalar multiplications in G_1 (to compute r_iP, $x_S H_1(m, U_i, Y_{R_i})$, and $r_i Y_{R_i}$) and $3n$ hash functions (to compute $H_1(m, U, Y_{R_i})$, $H_2(U, Y_{R_i}, rY_{R_i})$ and $H_3(U, Y_{R_i}, rY_{R_i})$). The ciphertext is $3n|G_1| + n|m|$. We summarize the above comparisons in the following Table 1. It is obvious that our scheme is more efficient than re-signcrypting a message n times using Yang et al.'s scheme.

Table 1: Efficiency comparison

	Scalar multiplications in G_1	Hash functions	Ciphertext size
RSY approach	$3n$	$3n$	$3n\|G_1\| + n\|m\|$
Our scheme	$n+2$	$n+2$	$(n+2)\|G_1\| + \|m\|$

6 Conclusions

We have proposed a multi-recipient signcryption scheme that broadcasts a message to multiple users in a secure and authenticated manner. Our scheme is proved to be secure in the random oracle model assuming the Gap Diffie-Hellman problem is computationally hard. Since it has much less computational costs and communication overheads than re-signcrypting a message n times using a signcryption scheme, we expect that our scheme can be used to transmit messages efficiently through the Internet.

References

[1] Zheng Y. Digital signcryption or how to achieve cost (signature & encryption) $\ll$ cost (signature) + cost(encryption). In: Advances in Cryptology-CRYPTO'97, LNCS 1294, Springer-Verlag, 1997. 165–179.

[2] Baek J, Steinfeld R, Zheng Y. Formal proofs for the security of signcryption. In: Public Key Cryptography-PKC'02, LNCS 2274, Springer-Verlag, 2002. 80–98.

[3] Malone-Lee J. Identity based signcryption. Cryptology ePrint Archive, Report 2002/098, 2002. Available from: http://eprint.iacr.org/2002/098.

[4] Libert B, Quisquater J J. A new identity based signcryption schemes from pairings. In: 2003 IEEE information theory workshop, Paris, France, 2003. 155–158.

[5] Chow S S M, Yiu S M, Hui L C K, et al. Efficient forward and provably secure ID-based signcryption scheme with public verifiability and public ciphertext authenticity. In: Information Security and Cryptology-ICISC'03, LNCS 2971, Springer-Verlag, 2004. 352–369.

[6] Boyen X. Multipurpose identity-based signcryption: a swiss army knife for identity-based cryptography. In: Advances in Cryptology-CRYPTO 2003, LNCS 2729, Springer-Verlag, 2003. 383–399.

[7] Yuen T H, Wei V K. Fast and proven secure blind identity-based signcryption from pairings. In: Topics in Cryptology-CT-RSA 2005, LNCS 3376, Springer-Verlag, 2005. 305–322.

[8] Chen L, Malone-Lee J. Improved identity-based signcryption. In: Public Key Cryptography-PKC 2005, LNCS 3386, Springer-Verlag, 2005. 362–379.

[9] Barreto P S L M, Libert B, McCullagh N, et al. Efficient and provably-secure identity-based signatures and signcryption from bilinear maps. In: Advances in Cryptology-ASIACRYPT 2005, LNCS 3788, Springer-Verlag, 2005. 515–532.

[10] Li X, Chen K. Identity based proxy signcryption scheme from pairings. In: 2004 IEEE International Conference on Services Computing, Shanghai, China, 2004. 494–497.

[11] Wang Q, Cao Z. Efficient ID-based proxy signature and proxy signcryption form bilinear pairings. In: Computational Intelligence and Security-CIS 2005, LNAI 3802, Springer-Verlag, 2005. 167–172.

[12] Duan S, Cao Z, Zhou Y. Secure delegation-by-warrant ID-based proxy signcryption scheme. In: Computational Intelligence and Security-CIS 2005, LNAI 3802, Springer-Verlag, 2005. 445–450.

[13] Duan S, Cao Z, Lu R. Robust ID-based threshold signcryption scheme from pairings. In: 2004 International Conference on Information Security, Shanghai, China, 2004. 33–37.

[14] Peng C, Li X. An identity-based threshold signcryption scheme with semantic security. In: Computational Intelligence and Security-CIS 2005, LNAI 3802, Springer-Verlag, 2005. 173–179.

[15] Huang X, Susilo W, Mu Y, Zhang F. Identity-based ring signcryption schemes: cryptographic prim-

itives for preserving privacy and authenticity in the ubiquitous world. In: Advanced Information Networking and Applications-AINA'05, Taipei, Taiwan, 2005. 649–654.

[16] Libert B, Quisquater J J. Efficient signcryption with key privacy from gap Diffie-Hellman groups. In: Public Key Cryptography-PKC 2004, LNCS 2947, Springer-Verlag, 2004. 187–200.

[17] Yang G, Wong D S, Deng X. Analysis and improvement of a signcryption scheme with key privacy. In: Information Security Conference-ISC 2005, LNCS 3650, Springer-Verlag, 2005. 218–232.

[18] Libert B, Quisquater J J. Improved signcryption from q-Diffie-Hellman problems. In: Security Communication Networks-SCN 2004, LNCS 3352, Springer-Verlag, 2005. 220–234.

[19] Ma C, Chen K, Zheng D, Liu S. Efficient and proactive threshold signcryption. In: Information Security Conference-ISC 2005, LNCS 3650, Springer-Verlag, 2005. 233–243.

[20] Zheng Y. Signcryption and its applications in efficient public key solutions. In: Information Security Workshop-ISW'97, LNCS 1396, Springer-Verlag, 1997. 291–312.

[21] Seo M, Kim K. Electronic funds transfer protocol using domain-verifiable signcryption scheme. In: Information Security and Cryptology-ICISC'99, LNCS 1787, Springer-Verlag, 1999. 269–277.

[22] Boneh D, Franklin M. Identity-based encryption from the weil pairing. In: Advances in Cryptology-CRYPTO 2001, LNCS 2139, Springer-Verlag, 2001. 213–229.

[23] Joux A, Nguyen K. Separating decision Diffie-Hellman from Diffie-Hellman in cryptographic groups. Journal of Cryptology, 2003, 16(4): 239–247.

[24] Gamage C, Leiwo J, Zheng Y. Encrypted message authentication by firewalls. In: Public Key Cryptography-PKC'99, LNCS 1560, Springer-Verlag, 1999. 69–81.