Available online at www.sciencedirect.com

SCIENCE DIRECT®

APPLIED MATHEMATICS AND COMPUTATION

ELSEVIER Applied Mathematics and Computation 160 (2005) 245–260

www.elsevier.com/locate/amc

New identity-based society oriented signature schemes from pairings on elliptic curves

Chih-Yin Lin [a], Tzong-Chen Wu [b,*], Fangguo Zhang [c], Jing-Jang Hwang [d]

[a] *Institute of Information Management, National Chiao Tung University, Hsinchu 300, Taiwan, ROC*

[b] *Department of Information Management, National Taiwan University of Science and Technology, 43, Section 4, Keelung Road, Taipei 106, Taiwan, ROC*

[c] *International Research Center for Information Security (IRIS), Information and Communications University, 58-4 Hwaam-dong Yusong-ku, Taejon 305-732, South Korea*

[d] *Department of Information Management, Chang Gung University, Kwei-Shan Tao-Yuan 333, Taiwan, ROC*

Abstract

In this paper, we will propose two identity-based society oriented signature schemes that allow a group of co-signers to collaboratively generate a single signature for a message. The first proposed scheme is designated with known signers and the second scheme is with anonymous signers. Both schemes make use of pairings on elliptic curves in construction and thus have the merits of simplicity in design and efficiency in performance. In the proposed scheme with anonymous signers, a signer may participate in several different signing groups and may join or leave a signing group dynamically in a secure and efficient manner.

Keywords: Cryptography; Digital signature; Identity-based; Society oriented; Multisignature; Pairing; Elliptic curve

[*] Corresponding author.
E-mail address: tcwu@cs.ntust.edu.tw (T.-C. Wu).

0096-3003/$ - see front matter © 2003 Elsevier Inc. All rights reserved.
doi:10.1016/j.amc.2003.07.016

1. Introduction

The concept of society oriented signature is first addressed by Desmedt [1], in which multiple signers collaboratively generate a single signature. There are two types of society oriented signature schemes: one is with known signers and the other is with anonymous signers [1,2]. The scheme with known signers is mostly referred to as multisignature schemes, e.g. the schemes in [3–6]. Within such schemes, the verifier makes use of the public keys from all co-signers for signature verification. In the scheme with anonymous signers, a group public key is established for the group of co-signers. Then, the verifier requires only this group public key to verify the society oriented signature. That is, the verifier does not necessarily know the identities of any co-signers, or how many co-signers have participated in signing.

Recently, Saeednia [2] proposed an identity-based society oriented signature scheme with anonymous signers, in which the signatures are verified with respect to only the group public identity. His scheme is converted from a well-known multisignature scheme proposed by Guillou and Quisquater [7]. Like the RSA cryptosystem, the security and arithmetic operations of their schemes are based on the factorization of a large composite. Saeednia's scheme also deals with situations that a signer may join or leave the signing group, or participate in different signing groups. As he claimed, the group public identity and all co-signers' private keys remain unaltered in such dynamic situations. However, his scheme fails to satisfy some more important security features when dynamic situations are considered. As specified by Wang and Zhu [8], the co-signer's private key may be disclosed and the signatures may be forged after dealing a dynamic situation in Saeednia's scheme. In addition, when all co-signers in the group leave, they can continue together to maliciously generate valid signatures and remain being anonymous. To withstand such coalition, Saeednia assumes that at least one co-signer should be honest. Nevertheless, this assumption does not help Saeednia's scheme to against Wang and Zhu's attacks.

In this paper, we will propose two identity-based society oriented signature schemes from pairings on elliptic curves. Our schemes make use of a recently proposed pairing-based identity-based signature scheme, i.e. Cha and Cheon's scheme [9], as the basic scheme. The Cha–Cheon scheme is provably secure against existential forgery on adaptively chosen message and identity attack in the random oracle model. Extended from their scheme, the first proposed scheme is named the SSK scheme that is designated to realize multisignatures with known signers. The second proposed scheme is named the SSA scheme, which is a multisignature scheme that achieves signer anonymity. In the SSA scheme, we will incorporate the concept of time or valid period to the group public key. Specifically, the society oriented signatures will be verified with respect to the group public identity along with a notion of valid time period. For example, the group public key can be the hashed digest of the group public identity concatenates the current date, i.e. hash(groupA||20030322). In this way, the verifier still makes use of the group public identity to verify society oriented signatures, despite of the necessity to reference which time period the signature was generated. As specified in [10,11], such time variant or time control approach can fit effectively and efficiently into pairing-based identity-based public keys. With this approach, our SSA scheme can effectively deals with dynamic situations while being immune to Wang and Zhu's attacks [8]. Moreover, the techniques we use herein can be applied to fix the flaws in Saeednia's scheme [2]. Details of secure dynamic situations will be discussed in Section 5.

Besides the merits of efficiency in computation and communication inherent from the underlying elliptic curve realization of pairings [10], the proposed schemes have the following characteristics:

(i) The size of the society oriented signature is fixed regardless of the number of co-signers.
(ii) The signature verification algorithms for the society oriented signature and the individual signature generated by the co-signer are the same as that in the Cha–Choen scheme.
(iii) Dynamic situations that a signer joins or leaves the signing group can be effectively and efficiently resolved, while the signatures are verified with respect to the same group identity.
(iv) Individual signer's private key remains unaltered for dynamic considerations.
(v) Any signer within the system can participate in several different signing groups with a single private key.
(vi) If all co-signers in the group leave, they can no longer generate valid signatures, even all of them collude.

The rest of this paper is organized as follows. In Section 2, we will address the properties of the pairing. In Section 3, we will review the Cha–Cheon's scheme. After that, details of the SSK and SSA schemes are specified in Section 4. In Section 5, we will discuss the security and the dynamic situations. Performance is also analyzed in Section 5, where we will show the exact computational costs for the SSK and SSA schemes. Finally, conclusions are given in Section 6.

2. The pairing

In the world of elliptic curve cryptography, the pairing was initially considered as a negative property. This is because it reduces the discrete logarithm problem on some elliptic curves (e.g., supersingular curves) to the discrete logarithm problem in a finite field [12], thus diminishing the strength and practicability of supersingular curves in cryptography. Until a tripartite key agreement protocol proposed by Joux in ANTS 2000 [13], the pairing for the first time became beneficial and favorable to cryptographic research and applications. Later, Boneh and Franklin [10] proposed an identity-based encryption scheme based on the modified Weil pairing and gave thorough analyses about its properties, security and performance. Since then, several pairing-based cryptographic schemes have been proposed, including a signature scheme [14], threshold signature, multisignature and blind signature schemes [3], etc., for general certificate-based public keys; and signature schemes [9,15], blind signature and ring signature schemes [16], etc, for identity-based public keys.

In this paper, we will follow most of the notations and parameters defined in [10]. Assume G_1 is an additive cyclic group of prime order q; and, G_2 is a multiplicative cyclic group of prime order q. The discrete logarithm problem in both G_1 and G_2 are hard. Usually, G_1 can be considered as a subgroup of points on an elliptic curve over a finite field; and, G_2 a subgroup of the multiplicative group of a related finite field. It is assumed herein that the decisional Diffie–Hellman problem is easy and the computational Diffie–Hellman problem is hard, which are defined as

Decision Diffie–Hellman—For $a, b, c \in Z_q^*$, given $P, aP, bP, cP \in G_1$, decide whether $c = ab$.
Computational Diffie–Hellman—For $a, b \in Z_q^*$, given $P, aP, bP \in G_1$, compute $abP \in G_1$.

We define pairing $e : G_1 \times G_1 \to G_2$ as the bilinear map that has the following properties:

(i) *Bilinear*: For all $P, Q \in G_1$ and $a, b \in Z_q^*$, we have $e(aP, bQ) = e(abP, Q) = e(P, abQ) = e(P, Q)^{ab}$.
(ii) *Non-degenerate*: There exists a $P \in G_1$, such that $e(P, P) \neq 1$.
(iii) *Computable*: Given $P, Q \in G_1$, there is an efficient algorithm to compute $e(P, Q)$.

Notice that property (iii) is supported by a polynomial time algorithm invented by Miller [17]. Much of the details of the parameter selection, efficiency and security analysis about pairings can be found in [10].

3. Review of Cha–Cheon's scheme

Assume there is a system administrator SA responsible for setting up the identity-based cryptosystem [18]. Let G_1 and G_2 be two cyclic groups of prime order q, as defined in Section 2. $P \in G_1$ is a public element satisfying $e(P,P) \neq 1$, i.e. P is a generator of G_1. SA's private key pair is $s \in Z_q^*$, and public key is $P_{\text{pub}} = sP$. Let $H_1 : \{0,1\}^* \to G_1$, $H_c : \{0,1\}^* \times G_1 \to Z_q$ be two cryptographic hash functions. Each signer u_i has an identity bit-string ID_i, which is uniquely determined form his name, address, etc. We define $H_1(\text{ID}_i)$ as u_i's public identity, which will be served as his public key. There are three algorithms in this scheme, i.e. *KeyGen*, *Sign* and *Verify*, respectively for key generation, signature generation and verification.

KeyGen. Given SA's private key s and signer u's public identity $H_1(\text{ID})$, compute u's private key is K as

$$K = sH_1(\text{ID}). \tag{1}$$

Sign. Given u's private key K and a message m, compute $R = rH_1(\text{ID})$, $h = H_c(m,R)$, and $S = (r+h)K$, where $r_i \in Z_q^*$ is randomly chosen. The signature is (S,R).

Verify. Given u's public identity $H_1(\text{ID})$, the message m and the signature (S,R), compute $h = H_c(m,R)$ and verify that if the following equation holds:

$$e(P,S) = e(P_{\text{pub}}, R + hH_1(\text{ID})). \tag{2}$$

4. Proposed schemes

By using the same identity-based setting in the Cha–Cheon scheme, we will propose two society oriented signature schemes, SSK and SSA. As previously specified, the SSK scheme realizes multisignatures with known signers that require public identities from all co-signers for signature verification. The SSA scheme realizes multisignatures with anonymous signers, in which the signatures are verified with respect to the group identity and the group current status. Both schemes employ the same system parameters and notations from the Cha–Cheon scheme.

4.1. The SSK scheme

There are five algorithms in this scheme, *SSKKeyGen*, *SSKIndSign*, *SSKIndVerify*, *SSKSigGen* and *SSKVerify*. *SSKKeyGen* is the same as that in the basic scheme. *SSKIndSig* and *SSKIndVerify* are used for individual signature generation. *SSKSigGen* and *SSKVerify* are used for society oriented signature generation and verification. Meanwhile, a clerk CLK is employed to collect individual signatures generated by co-signers and to construct the society oriented signature by the algorithm *SSKSigGen*. Note that CLK does not possess any secret information.

Without loss of generality, suppose n signers in the system will collaboratively generate a signature (S_Q, R_Q) for message m. We denote these signers as u_i's, for $i = 1, 2, \ldots, n$, who form a signing group Q. Each u_i initially computes his individual signature (S_i, R_i, R_Q) by *SSKIndSign*, and then sends it to CLK. Then, CLK verifies the received (S_i, R_i, R_Q) by *SSKIndVerify*. After CLK collects and verifies all individual signatures from u_i's, for $i = 1, 2, \ldots, n$, he constructs the society oriented signature (S_Q, R_Q) by *SSKSigGen*. The society oriented signature (S_Q, R_Q) can be publicly verified by *SSKVerify*, using the public identities $H_1(\mathrm{ID}_i)$'s from u_i's, for $i = 1, 2, \ldots, n$. Details of the these algorithms are stated as follows:

SSKKeyGen. The same as *KeyGen* in the Cha–Cheon scheme.

SSKIndSign. Given a message m, u_i compute the signature (S_i, R_i, R_Q) with other co-signers as follows.

Step 1. Select $r_i \in Z_q^*$ at random.

Step 2. Compute R_i and send it to other co-signers, where

$$R_i = r_i H_1(\mathrm{ID}_i). \tag{3}$$

Step 3. Compute R_Q and h as

$$R_Q = \sum_{i=1}^{n} R_i, \tag{4}$$

$$h = H_c(m, R_Q). \tag{5}$$

Step 4. Compute S_i as

$$S_i = (r_i + h)K_i. \tag{6}$$

SSKIndVerify. Given u_i's public identity $H_1(\mathrm{ID}_i)$ and u_i's individual signature (S_i, R_i, R_Q) on message m, compute $h = H_c(m, R_Q)$ and check if the following equation holds:

$$e(P, S_i) = e(P_{\mathrm{pub}}, R_i + hH_1(\mathrm{ID}_i)). \tag{7}$$

SSKSigGen. Given n co-signers' individual signatures (S_i, R_i, R_Q), for $i = 1, 2, \ldots, n$, on message m, ensure that all of them are valid, compute S_Q as below. The society oriented signature is (S_Q, R_Q).

$$S_Q = \sum_{i=1}^{n} S_i. \tag{8}$$

SSKVerify. Given u_i's public identities $H_1(\mathrm{ID}_i)$, for $i = 1, 2, \ldots, n$, the message m and the society oriented signature (S_Q, R_Q) collaboratively generated by all co-signers, compute $h = H_c(m, R_Q)$ and check if the following equation holds:

$$e(P, S_Q) = e\left(P_{\text{pub}}, R_Q + h\left(\sum_{i=1}^{n} H_1(\text{ID}_i)\right)\right). \tag{9}$$

Next, we will show the correctness of the SSK scheme in regard to the individual signature and the society oriented signature as follows:

Theorem 1. *In cooperation with all other co-signers, an honest co-signer u_i, who follows SSKIndSign, will generate an individual signature (S_i, R_i, R_Q) for message m that can be successfully verified by Eq.* (7) *in SSKIndVerify.*

Proof. By Eq. (5), we have $h = H_c(m, R_Q)$. Eq. (7) can be obtained by the following induction:

$$\begin{aligned} e(P, S_i) &= e(P, (r_i + h)K_i) \quad \text{(by Eq. (6))} \\ &= e(P, (r_i + h)sH_1(\text{ID}_i)) \quad \text{(byEq. (1))} \\ &= e(sP, (r_i + h)H_1(\text{ID}_i)) \quad \text{(bilinear property of } e) \\ &= e(P_{\text{pub}}, r_i D_i + hH_1(\text{ID}_i)) \quad \text{(bilinear property of } e) \\ &= e(P_{\text{pub}}, R_i + hH_1(\text{ID}_i)). \quad \text{(by Eq. (3))} \qquad \square \end{aligned}$$

Theorem 2. *If all co-signers honestly follow SSKIndSign and the trust CLK honestly follow SSKSigGen, the multisignature (S_Q, R_Q) can be successfully verified by the Eq.* (9) *in SSKVerify.*

Proof. By Eq. (5), we have $h = H_c(m, R_Q)$. Eq. (9) can be obtained by the following induction:

$$\begin{aligned} e(P, S_Q) &= e\left(P, \sum_{i=1}^{n} S_i\right) \quad \text{(by Eq. (4))} \\ &= \prod_{i=1}^{n} e(P, S_i) \quad \text{(bilinear property of } e) \\ &= \prod_{i=1}^{n} e(P_{\text{pub}}, R_i + hH_1(\text{ID}_i)) \quad \text{(by Theorem 1)} \\ &= e\left(P_{\text{pub}}, \sum_{i=1}^{n} R_i + h\left(\sum_{i=1}^{n} H_1(\text{ID}_i)\right)\right) \quad \text{(bilinear property of } e) \\ &= e\left(P_{\text{pub}}, R_G + h\left(\sum_{i=1}^{n} H_1(\text{ID}_i)\right)\right) \quad \text{(by Eq. (4))} \qquad \square \end{aligned}$$

4.2. The SSA scheme

Without loss of generality, assume a group of n signer $Q = \{u_1, u_2, \ldots, u_n\}$ who will cooperatively generate signatures. Let ID_Q be the unique group public identity of Q. We define the group public key for verifying Q's signature be $H_1(\mathrm{ID}_Q \| t)$, where t is a notion that indicates the time period. The time period should be defined according to the frequency of the dynamic situations. For example, a group with daily updates may employ $H_1(\mathrm{ID}_Q \| 20030213)$ as the group public key. For a group of more stable members, its group public key may use a longer period, e.g. $H_1(\mathrm{ID}_Q \| 200304\text{–}06)$. In this way, the verifier still verifies the society oriented signature with respect to the group public identity, despite that he has to reference when the signature was generated. In practice, the time period t should be predefined and publicly acknowledged. However, a more feasible way is to append m with the information of current t as the target to be signed. That is, m will be time-stamped before signed, according to when the signature is generated. Consequently, the verifier can always use the correct $H_1(\mathrm{ID}_Q \| t)$ to verify the signature, since t or the information that reveals t is available from the message to be verified.

Besides the use of time variant group public keys, we suppose all co-signers sign $m \| \mathrm{ID}_Q$ instead of m, in order to withstand Wang and Zhu's attacks [8]. Because only a concatenation is used, this simple approach will incur no extra cost, nor any security hazards.

There are six algorithms in the SSA scheme: *SSAKeyGen*, *SSAToken*, *SSAIndSign*, *SSAIndVerify*, *SSASigGen* and *SSAVerify*. *SSAKeyGen* is the same as that in the Cha–Cheon scheme. *SSAIndSign* and *SSAIndVerify* are individually the same as *SSKIndSign* and *SSKIndVerify* in the SSK scheme. To achieve individual u_i's anonymity in Q, SA will compute public group token T_Q with the *SSAToken* algorithm. A clerk CLK who is defined as in the SSK scheme will construct the society oriented signature with the group token T_Q. *SSASigGen* and *SSAVerify* are used respectively for society oriented signature generation and verification.

When generating the society oriented signature, each u_i will first computes his individual signature (S_i, R_i, R_Q) by *SSAIndSign*, and sends it to CLK. Then, CLK verifies the received (S_i, R_i, R_Q) by *SSAIndVerify*. After CLK collects and verifies individual signatures from all $u_i \in G$, he constructs the society oriented signature (S_Q, R_Q) by *SSKSigGen*. The society oriented signature (S_Q, R_Q) can be, afterwards, publicly verified by *SSKVerify*, with respect to the group public identity ID_Q and its current status. Details of these algorithms are stated as follows:

SSAKeyGen. The same as *KeyGen* in the Cha–Cheon scheme.

SSAToken. Given SA's private key s, group G's public identity ID_Q, current time period t, and co-signers' public identity ID_i's for all $u_i \in G$, compute T_Q by the equation below. T_Q is the group token for Q.

$$T_Q = s\left(H_1(\mathrm{ID}_Q\|t) - \left(\sum_{i=1}^{n} H_1(ID_i)\right)\right). \tag{10}$$

SSAIndSign. The same as *SSKIndSign* in the SSK scheme, except that $h = H_c(m\|\mathrm{ID}_Q, R_Q)$.

SSAIndVerify. The same as *SSKIndVerify* in the SSK scheme, except that $h = H_c(m\|\mathrm{ID}_Q, R_Q)$.

SSASigGen. Given message m and n co-signers' individual signatures (S_i, R_i, R_Q)'s, for $i = 1, 2, \ldots, n$, ensure that all (S_i, R_i, R_Q)'s are valid, compute $h = H_c(m\|\mathrm{ID}_Q, R_Q)$ and S_Q as in below. The society oriented signature is (S_Q, R_Q).

$$S_Q = \sum_{i=1}^{n} S_i + hT_Q. \tag{11}$$

SSAVerify. Given group Q's group public key $H_1(\mathrm{ID}_Q\|t)$, message m and the society oriented signature (S_Q, R_Q), compute $h = H_c(m\|\mathrm{ID}_Q, R_Q)$ and check if the following equation holds:

$$e(P, S_Q) = e(P_{\mathrm{pub}}, R_Q + hH_1(\mathrm{ID}_Q\|t)). \tag{12}$$

In the SSA scheme, each co-signer follows *SSAIndSign* to generate the individual signature, which is the same as *SSKIndSign*. Therefore, the correctness of the individual signature can be directly implied from Theorem 1. In the following, we will show the correctness of the society oriented signature in the SSA scheme:

Theorem 3. *If all co-signers honestly follow SSAIndSign and the trust CLK honestly follow SSASigGen, the multisignature* (S_Q, R_Q) *can be successfully verified by the Eq.* (12) *in SSAVerify.*

Proof. By Eq. (5), we have $h = H_c(m\|\mathrm{ID}_Q, R_Q)$. Eq. (12) can be obtained by the following induction:

$$\begin{aligned} e(P, S_Q) &= e\left(P, \left(\sum_{i=1}^{n} S_i\right) + hT_Q\right) \quad \text{(by Eq. (11))} \\ &= e\left(P, \left(\sum_{i=1}^{n} (r_i + h)K_i\right)\right. \\ &\qquad \left. + hs\left(H_1(\mathrm{ID}_Q\|t) - \sum_{i=1}^{n} H_1(\mathrm{ID}_i)\right)\right) \quad \text{(by Eqs. (6) and (10))} \end{aligned}$$

$$
\begin{aligned}
&= e\left(P, \left(\sum_{i=1}^{n}(r_i + h)sH_1(\mathrm{ID}_i)\right)\right. \\
&\qquad \left. + hs\left(H_1(\mathrm{ID}_Q \| t) - \sum_{i=1}^{n} H_1(\mathrm{ID}_i)\right)\right) \quad \text{(by Eq. (1))} \\
&= e\left(P, s\left(\left(\sum_{i=1}^{n} r_i H_1(\mathrm{ID}_i)\right) + \sum_{i=1}^{n} hH_1(\mathrm{ID}_i)\right.\right. \\
&\qquad \left.\left. + h\left(H_1(\mathrm{ID}_Q \| t) - \sum_{i=1}^{n} H_1(\mathrm{ID}_i)\right)\right)\right) \\
&= e\left(sP, \sum_{i=1}^{n} r_i H_1(\mathrm{ID}_i) + h\left(\left(\sum_{i=1}^{n} H_1(\mathrm{ID}_i)\right)\right.\right. \\
&\qquad \left.\left. + H_1(\mathrm{ID}_Q \| t) - \left(\sum_{i=1}^{n} H_1(\mathrm{ID}_i)\right)\right)\right) \quad \text{(bilinear property of } e\text{)} \\
&= e\left(P_{\mathrm{pub}}, \sum_{i=1}^{n} R_i + hH_1(\mathrm{ID}_Q \| t)\right) \quad \text{(by Eq. (3))} \\
&= e(P_{\mathrm{pub}}, R_Q + hH_1(\mathrm{ID}_Q \| t)) \quad \text{(by Eq. (4))} \qquad \square
\end{aligned}
$$

5. Discussions

In Section 5.1, we discuss the security of the proposed schemes without considering the dynamic situations. How to deal with dynamic situation and the possible security concerns will be given in Section 5.2. Then, we will analyze the performance of the proposed SSK and SSA schemes in Section 5.3.

5.1. Security

We consider two types of attacks to both proposed schemes: the outsider forgery attack [19] and the insider forgery attack [3,9,19]. In specific to the SSA scheme, we further consider that the adversary may try to disclose the identity of any participant signer to violate the signer anonymity property, i.e. the anti-anonymity attack. The definitions of these attacks are given as follows:

Outsider forgery attack—An adversary A, who is not in the signing group Q as a co-signer, i.e. $A \notin Q$, may attempt to for a society oriented signature for a chosen message. In this attack, we assume that all public information is available to A.

Insider forgery attack—A co-signer in the signing group or the collusion of some co-signers may attempt to forge the multisignature for the signing group, under the assumption that all public information is available to colluded co-signers. In this attack, we follow the same scenario in analyzing the insider attack against most multisignature schemes [3,19] by assuming the number of malicious co-signers in Q can be as many as $n-1$.

Anti-anonymity attack—An adversary A, who is not in the signing group Q as a co-signer, i.e. $A \notin Q$, attempts to disclose the identities of the participant co-signers from a society oriented signature under the assumption that all public information is available to A. Note that in the SSA scheme, we assume all individual signatures are private to co-signers and the CLK, and none of them will deliberately reveal any co-signer's identity information to outsiders.

Due to the employment as the basic scheme, the security of the proposed schemes is based on the robustness of the Cha–Cheon scheme. Besides, we assume that all one-way hash functions used herein are secure for cryptographic usages, as those defined in [20]. Below, we will prove the security of the individual signature, and then show that the proposed SSK and SSA schemes are secure against the above attacks.

Theorem 4. *The security of the individual signature is equivalent to the signature in the Cha–Cheon scheme under the assumption that the one-way hash function* H_c *is secure.*

Proof. In the Cha–Cheon scheme, a valid signature for message m is (S, R), and its verification equation (2) can be represented as

$$e(P,S) = e(P_{\text{pub}}, R + H_c(m,R)H_1(\text{ID})).$$

In the proposed SSK scheme, a valid individual signature for message m is (S_i, R_i, R_Q), where $R_Q = \sum_{i=1}^{n} R_i$. The verification equation (7) can be represented as

$$e(P,S_i) = e(P_{\text{pub}}, R_i + H_c(m, R_i + R_\Sigma)H_1(\text{ID}_i)), \tag{13}$$

where $R_\Sigma = R_Q - R_i = \sum_{j=1,j\neq i}^{n} R_j$.

In Eq. (13), if R_Σ is fixed in advance, then the construction of Eq. (13) is related to Eq. (12), which implies that finding a valid signature (S_i, R_i) for Eq. (13) will require the same knowledge as the case for Eq. (12). On the other hand, if S_i is fixed prior to the computing of (R_i, R_Σ) to satisfy Eq. (13), the adversary will have to convert H_c to attempting this. Under the assumption that H_c is a secure one-way hash function, the security of the individual signature in the proposed schemes is equivalent to that of the signature in the Cha–Cheon scheme, which is secure against adaptively chosen message and identity attack in the random oracle model [9]. The same result can be obtained in the SSA scheme; since the only difference is that m is replaced by $m\|\mathrm{ID}_Q$ in the one-way hash function. $\square$

Theorem 5. *The SSK scheme is secure against the outsider forgery attack and the insider forgery attack.*

Proof. For the outsider forgery attack, consider that an adversary $A \notin Q$ wants to forge the multisignature of some message m for all $u_i \in Q$. That is, A knows all public information, including the public identities $H_1(\mathrm{ID}_i)$'s for all $u_i \in Q$, and wants to find (S_Q, R_Q) satisfying the verification equation (9) in *SSKVerify*. By letting the public verification key for Q as $\sum_{i=1}^{n} H_1(\mathrm{ID}_i)$, the construction of the multisignature and the multisignature verification of the SSK scheme can be related to the signature and the *Verify* algorithm in the Cha–Cheon scheme. This implies that such attack is equivalent to the signature forgery in their scheme. Since the Cha–Cheon scheme is secure against existential forgery on adaptively chosen message and identity attack in the random oracle model [9], the outsider forgery attack is infeasible in the proposed SSK scheme.

For the insider forgery attack, we assume there is at least one honest co-signer u_a in Q. Considering that some malicious signers u_j's, for $u_j \in Q \setminus \{u_a\}$, who want to generate the multisignature of the message m for the signing group Q. From *SSKSigGen*, it is to see that all malicious co-signers have to obtain u_a's individual signature to attempt this. With all public information and individual signatures generated by u_a regarding some messages different to m, all $u_j \in Q \setminus \{u_a\}$ may try to deduce $u_a a$'s private key or forge u_a's individual signature for m. However, deducing u_a's private key $K_a = sH_1(\mathrm{ID}_a)$ from his public key $H_1(\mathrm{ID}_a)$ requires the knowledge of SA's private key s, and finding s from SA's public key $P_{\mathrm{pub}} = sP$ is a problem of solving discrete logarithm in G_1, which is widely believed to be computationally infeasible if G_1 is well-chosen [12,21]. On the other hand, the individual signature (R_Q, R_a, S_a) has the same security strength as the signature in the Cha–Cheon scheme, as proved from Theorem 4. The insider forgery attack is infeasible. $\square$

Theorem 6. *The SSA scheme is secure against the outsider attack, the insider attack and the anti-anonymity attack.*

Proof. The security of the SSA scheme regarding the outsider forgery attack and the insider forgery attack can be directly implied from Theorem 5.

For the anti-anonymity attack, an adversary may obtain message m with its society oriented signature (R_Q, S_Q), and uses the group public key $H_1(\mathrm{ID}_Q\|t)$ to verify its validity, as in Eq. (12). If ID_Q reveals no personal information of any co-signers in Q, the only public information that could possibly relate the group public key to the co-signers' individual public keys is the group token T_Q. If the adversary possesses all individual signatures (R_Q, R_i, S_i)'s and the society oriented signature (R_Q, S_Q) regarding the same message, he can relate them by Eq. (11). Moreover, he can compare R_Q in (R_Q, R_i, S_i) to R_Q in (R_Q, S_Q) to identify individual co-signer in Q. However, under the assumption that all individual signatures are private to co-signers and CLK, and none of them will deliberately reveal any signer's identity information to outsiders, this attack is infeasible. □

5.2. Dynamic situations

In the proposed SSA scheme, a signer may dynamically join or leave the signing group. To resolve this, SA simply computes a new group token $T_{Q'}$ for group Q', where Q' is the updated group after any signer joins or leaves the original group Q. Note that the group public identity information remains the same, i.e. $\mathrm{ID}_{Q'} = \mathrm{ID}_Q$. While computing the new group token, the group public key is also updated. For example, the group token is updated from $T_Q = s(H_1(\mathrm{ID}_Q \| 200303) - \sum_{u_i \in Q} H_1(\mathrm{ID}_i))$ to $T_{Q'} = s(H_1(\mathrm{ID}_{Q'} \| 200304) - \sum_{u_i \in Q'} H_1(\mathrm{ID}_i))$. As a result, the SSA scheme can effectively and efficiently deal with dynamic situations, since only the group public token is modified.

Regarding the flaws in Saeednia's scheme [2], Wang and Zhu [8] specified five scenarios that may expose security problems. These scenarios are: (i) a signer leaves the signing group, (ii) a signer joins the signing group, (iii) different groups represent the same organization, (iv) the same group represent different organizations, and (v) all signers leave the signing group and collude. In Saeednia's scheme, the difference of a new group token and an old one directly reveals some crucial information τ. In (i) and (ii), τ is the co-signer's private key who joins/leaves. In (iii), τ may imply the ratio of the private keys of two co-signers u_a and u_b, thus forging an individual signature $(S_\mathrm{a}, R_\mathrm{a}, R_Q)$ from another valid signature $(S_\mathrm{a}, R_\mathrm{a}, R_Q)$ is possible. In (iv), τ may indicate ratio of the private keys corresponding to two group public keys $H_1(\mathrm{ID}_{Q_1})$ and $H_1(\mathrm{ID}_{Q_2})$, thus forging a society oriented signature (S_{Q_1}, R_{Q_1}) from another valid signature (S_{Q_2}, R_{Q_2}) is possible. In (v), without the assumption of an honest signer, if all signers leave they can collude to sign maliciously and anonymously. As analyzed by Wang and Zhu [8], Saeednia's scheme can be enhanced to withstand attacks in scenarios (i), (ii), (iii) and (iv) if a trusted *CLK* exists who secretly possesses the group token.

In the SSA scheme, although both the new group token $T_{Q'}$ and the old one T_Q are derived from the same group public identity information, they are computed based on different time periods. Therefore, the difference of group tokens reveals no useful information to help malicious outsiders in scenario (i), (ii), and (iii). In (iv), although the difference of two different group tokens may imply the difference of the private keys corresponding to two different group public keys, a valid society oriented signature is embedded with group public identity with the message in H_c, i.e. $h = H_\mathrm{c}(m \| \mathrm{ID}_Q, R_Q)$. Therefore, to conduct

Table 1
Computational costs of the proposed SSK and SSA schemes (n signers)

	SSK	SSA
IndSign	$\mathrm{TH} + \mathrm{TA}_q + (n-1)\mathrm{TA}_{G_1} + 2\mathrm{TM}_{G_1}$	$\mathrm{TH} + \mathrm{TA}_q + (n-1)\mathrm{TA}_{G_1} + 2\mathrm{TM}_{G_1}$
IndVerify	$\mathrm{TH} + \mathrm{TA}_{G_1} + \mathrm{TM}_{G_1} + 2\mathrm{TP}$	$\mathrm{TH} + \mathrm{TA}_{G_1} + \mathrm{TM}_{G_1} + 2\mathrm{TP}$
SSAToken	–	$(n-1)\mathrm{TA}_{G_1} + \mathrm{TR}_{G_1} + \mathrm{TM}_{G_1}$
SSKSigGen/ SSASigGen	$\mathrm{TH} + (n-1)\mathrm{TA}_{G_1}$	$\mathrm{TH} + n\mathrm{TA}_{G_1} + \mathrm{TM}_{G_1}$
SSKVerify/ SSAVerify	$\mathrm{TH} + n\mathrm{TA}_{G_1} + \mathrm{TM}_{G_1} + 2\mathrm{TP}$	$\mathrm{TH} + \mathrm{TA}_{G_1} + \mathrm{TM}_{G_1} + 2\mathrm{TP}$

a successful forgery requires the ability to invert the one-way hash function H_c. In (v), when all signer leave, the time period also changes, therefore the coalition of them would not be able to generate valid society oriented signatures not belong to the same period. Due to the time variant group public key and the concatenation of message with group public identity when signing, the proposed SSA scheme is secure against Wang and Zhu's attacks in all scenarios without employing a trusted CLK.

5.3. Performance

The performance for computational efficiency is analyzed herein. We omit the cost for computing the hashed digest from ID_i to $H_1(\mathrm{ID}_i)$. The cost is measured in terms of the following arithmetic operations. The computational costs for the SSK and SSA schemes are given in Table 1, where n is the number of co-signers.

TH The time for computing the one-way hash function H_c.
TA_q The time for computing a modular addition Z_q.
TA_{G_1} The time for computing an addition in G_1.
TR_{G_1} The time for computing a subtraction in G_1.
TM_{G_1} The time for computing a multiplication in G_1.
TP The time for computing a pairing e.

6. Conclusions

In this paper, we have proposed two identity-based society oriented signature schemes, i.e. SSK and SSA, to respectively realize the multisignature scheme with known signers and with anonymous signers. We have shown that the proposed schemes work correctly and are secure under possible outsider and insider forgery attacks. As discussed, an attempt to disclose the signer anonymity in the SSA scheme is also infeasible. Due to the underlying pairing structure, the proposed schemes have the merits of simplicity in construction and efficiency in performance.

In the SSA scheme, we have provided an efficient and effective method to deal with dynamic situations that allow any signer to dynamically join or leave different signing groups with a single private key. By using the time variant group public key and signing the hash of message concatenates the group public identity, the proposed SSA scheme is secure against Wang and Zhu's attacks in all scenarios without requiring a trusted CLK.

References

[1] Y. Desmedt, Society and group oriented cryptography: a new concept, in: Advances in Cryptology—CRYPTO 87, Spring-Verlag, 1988, pp. 120–127.

[2] S. Saeednia, An identity-based society oriented signature scheme with anonymous signers, Inform. Process. Lett. 83 (2002) 295–299.

[3] A. Boldyreva, Efficient threshold signature, multisignature and blind signature schemes based on the gap-Diffie–Hellman-group signature scheme, in: Proceedings of International Workshop on Practice and Theory in Public Key Cryptography—PKC2003, Springer-Verlag, 2003, pp. 31–46.

[4] C. Boyd, Digital multisignatures, in: Proceedings of IMA Conference on Cryptography and Coding, Oxford University Press, 1989, pp. 241–246.

[5] T. Hardjono, Y. Zheng, A practical digital multisignature scheme based on discrete logarithms, in: Advance in Cryptology—AUSCRYPT 92, Springer-Verlag, 1992, pp. 122–132.

[6] S. Micali, K. Ohta, L. Reyzin, Accountable subgroup multisignatures, in: Proceedings of 8th ACM Conference on Computer and Communication Security, ACM press, 2001, pp. 245–254.

[7] L. Guillou, J.J. Quisquater, A paradoxical identity-based signature scheme resulting from zero-knowledge, in: Advances in Cryptology—CRYPTO 88, Springer-Verlag, 1989, pp. 216–231.

[8] G. Wang, B. Zhu, Remarks in Saeednia's identity-based society oriented signature scheme with anonymous signers, Cryptology ePrint Archive, Report 2003/046, 10 March 2003. Available from http://eprint.iacr.org/2003/046.

[9] J.C. Cha, J.H. Cheon, An identity-based signature from gap Diffie–Hellman groups, in: Proceedings of International Workshop on Practice and Theory in Public Key Cryptography—PKC 2003, Springer-Verlag, 2003, pp. 18–30.

[10] D. Boneh, M. Franklin, Identity-based encryption from the Weil pairing, in: Advances in Cryptology—CRYOTO 2001, Springer-Verlag, 2001, pp. 213–229.

[11] K.G. Paterson, Cryptography from pairings: a snapshot of current research, Information Security Technical Report 7 (3) (2002) 41–54. Available from http://www.isg.rhul.ac.uk/~kp/.

[12] A.J. Menezes, T. Okamoto, S. Vanstone, Reducing elliptic curve logarithms to logarithms in a finite field, IEEE Trans. Inform. Theory 39 (1993) 1639–1646.

[13] A. Joux, A one round protocol for tripartite Diffie–Hellman, in: Algorithmic Number Theory Symposium, ANTS-IV, Springer-Verlag, 2000, pp. 385–394.

[14] D. Boneh, H. Shacham, B. Lynn, Short signatures from the Weil pairing, in: Advances in Cryptology—AISACRYPT 2001, Springer-Verlag, 2001, pp. 514–532.

[15] K.G. Paterson, ID-based signatures from pairings on elliptic curves, Electron. Lett. 38 (18) (2002) 1025–1026.

[16] F. Zhang, K. Kim, ID-based blind signature and ring signature from pairings, in: Advances in Cryptology—ASIACRYPT 2002, Springer-Verlag, 2002, pp. 533–547.

[17] V. Miller, Short programs for functions on curves, unpublished manuscript, 1986.

[18] A. Shamir, Identity-based cryptosystems and signature schemes, in: Advances in Cryptology—CRYPTO 84, Springer-Verlag, 1984, pp. 47–53.

[19] M. Michels, P. Horster, On the risk of disruption in several multiparty signature schemes, in: Advances in Cryptology—ASIACRYPT 96, Springer-Verlag, 1996, pp. 334–345.

[20] NIST, Federal Information Processing Standard Publication 180-2, Secure Hash Standard (SHS), 2002. Available from http://csrc.nist.gov/publications/fips/fips180-2/fips180-2.pdf.

[21] IEEE, IEEE P1363 Draft Standard, Annex—A: Number Theoretic Algorithms, 1998. Available from http://grouper.ieee.org/groups/1363/.