Certificateless short sequential and broadcast multisignature schemes using elliptic curve bilinear pairings

SK Hafizul Islam *, G.P. Biswas

Department of Computer Science and Engineering, Indian School of Mines, Dhanbad 826004, Jharkhand, India

Received 8 July 2012; revised 27 December 2012; accepted 26 May 2013
Available online 31 May 2013

KEYWORDS
Public key infrastructure; Identity-based cryptography; Certificateless cryptography; Elliptic curve cryptography; Bilinear pairing; Short multisignature

Abstract Several certificateless short signature and multisignature schemes based on traditional public key infrastructure (PKI) or identity-based cryptosystem (IBC) have been proposed in the literature; however, no certificateless short sequential (or serial) multisignature (CL-SSMS) or short broadcast (or parallel) multisignature (CL-SBMS) schemes have been proposed. In this paper, we propose two such new CL-SSMS and CL-SBMS schemes based on elliptic curve bilinear pairing. Like any certificateless public key cryptosystem (CL-PKC), the proposed schemes are free from the public key certificate management burden and the private key escrow problem as found in PKI- and IBC-based cryptosystems, respectively. In addition, the requirements of the expected security level and the fixed length signature with constant verification time have been achieved in our schemes. The schemes are communication efficient as the length of the multisignature is equivalent to a single elliptic curve point and thus become the shortest possible multisignature scheme. The proposed schemes are then suitable for communication systems having resource constrained devices such as PDAs, mobile phones, RFID chips, and sensors where the communication bandwidth, battery life, computing power and storage space are limited.

1. Introduction

Digital signatures play a vital role in the security of information and communication networks by providing message integrity, authentication and non-repudiation during transmission over any insecure or hostile network. The property of message integrity guarantees that the receiver detects any alteration of the message during transmission, and the authentication property ensures the message generation by an expected sender. Compared with these two properties, the non-repudiation property is equally important, which assures that after creating a signature, the signer cannot deny the signature generation at a later time. However, in some real-life applications, such as electronic check signing, electronic contracts, decision-making processes, petitions, and workflow systems a message needs to be authenticated or approved by two or more persons concurrently. In this situation, a multisignature approach is more appropriate than any ordinary signature scheme. There are different multisignature schemes (Itakura and Nakamura, 1983; Harn, 1994; Chen and Hwang, 1994; Pon et al., 2002; Chen et al., 2004; Meng et al., 2007; Shim, 2008; Chang et al.,...
where two or more signers mutually sign on the same message to generate a single and valid multisignature. At a later time, the multisignature can be verified by a public verifier using the public keys of all the signers.

1.1. Literature review

Based on an extended RSA technique, Itakura and Nakamura (1983) first proposed a sequential (or serial) multisignature scheme, and other similar schemes are presented in (Pon et al., 2002; Meng et al., 2007; Gangishteti et al., 2006; Shim, 2008; Chu and Zhao, 2008). The CL-SSMS has many real-life applications such as when an electronic check needs to be signed serially by the various persons in an office based on their designation. On the other hand, the broadcast (or parallel) multisignature schemes can be found in (Harn and Ren, 2010; Chen et al., 2004; Chang et al., 2009; Harn, 1994; Chan and Hwang, 1994; Giri and Srivastava, 2007; Yang et al., 2010; Gui and Zhang, 2010). The multisignature schemes (Giri and Srivastava, 2007; Chu and Zhao, 2008; Le and Gabillon, 2009) designed upon traditional public key infrastructure (PKI) (Diffie and Hellman, 1976) have some problems such as the requirement of huge storage space to store the public key certificates, complicated management strategy to distribute the certificates and additional computing power to verify the certificates (Giri and Srivastava 2007; Chu and Zhao, 2008; Le and Gabillon, 2009; Das et al., 2013). The identity-based cryptosystem (IBC), first introduced by Shamir (1984), can solve these drawbacks because IBC abolishes the need for public key certificate management and distribution infrastructure (Gangishteti et al., 2006; Biao et al. 2010; Yang et al., 2010; Islam and Biswas 2013b, 2013c) as required in PKI. A user can derive his public key from a known identity such as an email address, and IP address and the public key can be revoked easily by just binding a time duration to it (Boneh and Franklin, 2001). However, because a trusted third party called the private key generator (PKG) is required to compute the corresponding private key, IBC becomes vulnerable to the private key escrow problem. To remove the key escrow problem of IBC, Al-Riyami and Paterson (2003) proposed the concept of certificateless public key cryptography (CL-PKC), where the PKG generates the identity-based partial private key and a user himself generates the full private key by using the partial private key received from PKG and his own chosen random secret vector. The PKG does not have access to the user’s full private key and hence, the private key escrow problem and the need for a public key certificate are solved in the CL-PKC system.

1.2. Motivations and contributions

Recently, the certificateless short signature (CL-SS) schemes (Huang et al., 2007; Chen et al., 2008; Du and Wen, 2009; Choi et al., 2011) have been used extensively in many resource-constrained wireless devices such as PDAs, mobile phones, RFID chips, and sensors where the communication bandwidth, battery life, computing power and storage space are limited. The short signature designed based on elliptic curve cryptography (ECC) can also offer high levels of security with comparatively short length signatures, and hence, most of the schemes use ECC (Miller, 1985; Koblitz, 1987) for the implementation of public key cryptosystems (PKC). Compared with other PKCs, the ECC-based PKC offers the same level of security with reduced key size, faster computation as well as less memory, energy and bandwidth usage, and thus, it is more suitable for resource-constrained devices. In the literature, several digital multisignature schemes (Itakura and Nakamura, 1983; Harn, 1994; Chen and Hwang, 1994; Pon et al., 2002; Chen et al., 2004; Gangishteti et al., 2006; Meng et al., 2007; Giri and Srivastava, 2007; Chu and Zhao, 2008; Shim, 2008; Chang et al., 2009; Le and Gabillon, 2009; Harn and Ren, 2010; Biao et al., 2010; Yang et al., 2010; Gui and Zhang, 2010) in PKI or IBC and many certificateless short signature schemes have been proposed; however, no certificateless short multisignature scheme has yet been designed. We combined the advantages of short signature and multisignature together with the features of CL-PKC and propose two efficient certificateless short sequential multisignature (CL-SSMS) and certificateless short broadcast multisignature (CL-SBMS) schemes using elliptic curve bilinear pairing (Boneh and Franklin, 2001). It is shown that both the schemes are secure and more computationally efficient than the others. The length of the proposed multisignature in both of the schemes is equal to an elliptic curve point and thus efficient in communication.

1.3. Paper organization

The rest of the paper is organized as follows. Section 2 describes some preliminary ideas about elliptic curve bilinear pairing and the related intractable hard problems. In Section 3, the two proposed certificateless short multisignature schemes CL-SSMS and CL-SBMS are described. The security and efficiency analyses of the schemes are given in Section 4, and Section 5 concludes the paper.

2. Preliminaries

This section briefly describes the basic concepts and properties of bilinear pairing and some computational hard problems, which are incorporated in our proposed signature schemes for achieving the desired security.

2.1. Bilinear pairing

Let \( G_q \) be an additive cyclic group of elliptic curve points of prime order \( q \) (where \( q \geq 2^k \) and \( k \) is security parameter) and \( G_m \) be a multiplicative group of the same order \( q \). Let \( \hat{e} : G_q \times G_q \rightarrow G_m \) be an admissible bilinear mapping that satisfies the following properties:

- **Bilinearity:** For any \( P, Q, R \in G_q \) then \( \hat{e}(P + Q, R) = \hat{e}(P, R)\hat{e}(Q, R) \) and \( \hat{e}(P, Q + R) = \hat{e}(P, Q)\hat{e}(P, R) \). Therefore, for any \( a, b \in \mathbb{Z}_q^* \) : \( \hat{e}(ap, bQ) = \hat{e}(P, Q)^ab = \hat{e}(abP, Q) = \hat{e}(P, abQ) \) holds.
- **Non-degeneracy:** There exists \( P, Q \in G_q \) such that \( \hat{e}(P, Q) \neq 1_m \), where \( 1_m \) is an identity element of \( G_m \).
- **Computability:** There must be an efficient algorithm, which can compute \( \hat{e}(P, Q) \) for all \( P, Q \in G_q \).

In general, \( G_q \) is a group of points on an elliptic curve and \( G_m \) is a multiplicative subgroup of a finite field. The bilinear
map \( \hat{\cdot} \) will be derived either from the modified Weil pairing or from the Tate pairing over a finite field. A more comprehensive description about bilinear pairings, selection of elliptic curves and suitable parameters and group formation can be found in (Boneh and Franklin, 2001; Boneh et al., 2004).

2.2. Computational problems

Some computational problems in the elliptic curve group and bilinear pairing, which are assumed to be secure and cannot be breached using a polynomial time-bounded algorithm (Koblitz, 1989; Silverman and Suzuki, 1998; Menezes et al., 1993; Frey et al., 1999; Gaudry, 2000), are described below.

- **Elliptic curve discrete logarithm \( \hat{\cdot} \) problem (ECDLP).** Given a random instance of \( P, Q \in G_q \), find an integer \( a \in \mathbb{Z}_q^* \) such that \( Q = aP \).

- **Computational Diffie-Hellman problem (CDHP):** Given a random instance of \( (P, aP, bP) \) for any \( a, b \in \mathbb{Z}_q^* \), the computation of \( abP \) is hard to the group \( G_q \).

- **Bilinear Diffie-Hellman problem (BDHP):** Given a random instance of \( (P, aP, bP, cP) \) and for any \( a, b, c \in \mathbb{Z}_q^* \), it is impossible to compute \( \hat{e}(P, Q)^{abc} \).

3. Proposed CL-SSMS and CL-SBMS schemes

In this section, two efficient short multisignature schemes, called certificateless short sequential or serial multisignature (CL-SSMS) and certificateless short broadcast or parallel multisignature (CL-SBMS) based on ECC and bilinear pairing are proposed. Let \( A = \{A_1, A_2, \ldots, A_n\} \) be a set of \( n \) signers and their respective identities \( ID = \{ID_1, ID_2, \ldots, ID_n\} \). Now each signer \( A_i (1 \leq i \leq n) \) generates full private key \( sk_i = (D_i, x_i) \) and public key \( pk_i = (Q_i, P_i) \) using the proposed scheme as described below.

3.1. Proposed CL-SSMS scheme

The signing order in our scheme is determined by either the message issuer or the signers themselves; however, the order is random. At the beginning, the message issuer issues and sends a message \( m \) (say) to \( A_1 \) as the first signer. Then \( A_1 \) computes the signature \( \sigma_1 \) on \( m \) and sends \( m, \sigma_1 \) to the next signer \( A_2 \). Upon receiving \( (m, \sigma_1) \), \( A_2 \) verifies \( (m, \sigma_1) \) and computes the signature \( (m, \sigma_2) \), and sends the same to the third signer \( A_3 \) for further signing. This process continues until the last signer signs the message \( m \). Thus, we can say that the signer \( A_i \) verifies \( (m, \sigma_{i-1}) \) received from the signer \( A_{i-1} \) and then produces the signature \( (m, \sigma_i) \) using his full private key. Finally, the last signer \( A_n \) generates the full multisignature with respect to all signers, which is allowed to be verified by any public verifier using the public keys of all the signers. The proposed CL-SSMS scheme consists of seven phases, the details of which are described below in the accompanying flow diagram as illustrated in Fig. 1.

- **Setup:** The PKG runs this algorithm to generate the system’s parameter. For a given security parameter \( k \in \mathbb{Z}^+ \), PKG does the following:

  (a) Choose an additive cyclic elliptic curve group \( (G_q, +) \) of prime order \( q \), a multiplicative group \( (G_m, \cdot) \) of order \( q \) and an admissible bilinear map \( \hat{\cdot}: G_q \times G_q \to G_m \).

  (b) Choose a number \( s \in \mathbb{Z}_q^* \), a generator point \( P \) of \( G_q \) and compute \( P_0 = sP \), where the private–public key pair of PKG is \( (s, P_0) \).

  (c) Choose two one-way and secure cryptographic hash functions \( H_1: [0, 1]^* \to G_q \) and \( H_2: [0, 1]^* \to G_m \).

  (d) Publish \( \Omega = \{G_q, G_m, \hat{\cdot}, q, P, P_0, H_1, H_2\} \) as the system’s parameter while the master key \( msk = s \) is kept secret by the PKG.

- **Set-Secret-Value:** The user \( ID \) picks a number \( x_i \in \mathbb{Z}_q^* \) as his secret value and then computes the corresponding public key as \( P_i = sP \).

- **Partial-private-key-extract:** This algorithm is executed by the PKG to generate users’ identity-based partial private keys. It takes \( \Omega \), master key \( msk = s \), user identity \( ID_i \) and partial public key \( P_i \) of \( ID_i \) as inputs and generates the partial private key \( D_i \) for \( ID_i \) as follows:

  (a) Compute \( Q_i = H_i(ID_i, P_i) \).

---

**Figure 1** The proposed CL-SSMS scheme.
(b) Compute the partial private key $D_i = sQ_i$ and send it to $ID_i$ via a secure channel.

- **Set-private-key:** The user $ID_i$ sets $sk_i = (D_i, x_i)$ as his full private key.

- **Set-public-key:** The user $ID_i$ sets $pk_i = (Q_i, P_i)$ as his full public key.

- **CL-SSMS-sign:** In order to generate a sequential short multisignature for a given message $m \in \{0, 1\}^*$, each signer $A_i (1 \leq i \leq n)$ performs the following operations:

  **Step 1:** The signer $A_1$
  (a) Computes $\sigma_1 = x_1H_2(m) + D_1$.
  (b) Sends the message-signature pair $(m, \sigma_1)$ to the next signer $A_2$.

  **Step 2:** The signer $A_2$
  (a) Verifies $(m, \sigma_1)$ by determining whether the equation $\hat{e}(\sigma_1, P) = \hat{e}(H_2(m), \sum_{i=1}^{n-1} P_i)$ holds.
  (b) If it holds, $A_2$ computes $\sigma_2 = \sigma_1 + x_2H_2(m) + D_2$ i.e., $\sigma_2 = x_1H_2(m) + x_2H_2(m) + D_1 + D_2$ and then sends $(m, \sigma_2)$ to the signer $A_3$

  Similarly, the signer $A_2$ signs and sends to $A_3$ and so on up to $A_{n-2}$ to $A_{n-1}$. All sequentially compute their signatures and complete the multisignature process.

  **Step n:** The last signer $A_n$
  (a) Verifies $(m, \sigma_{n-1})$ received from $A_{n-1}$ by determining whether the equation $\hat{e}(\sigma_{n-1}, P) = \hat{e}(H_2(m), \sum_{i=1}^{n-1} P_i)$ holds.
  (b) If it holds, $A_n$ computes $\sigma_n = \sigma_{n-1} + x_nH_2(m) + D_n$ i.e., $\sigma_n = \sum_{i=1}^{n-1} [x_iH_2(m) + D_i]$ (say) and then sends the final signature $(m, \sigma)$ to the verifier for verification.

- **CL-SSMS-verify:** In order to verify $(m, \sigma)$, the following steps are to be executed by the verifier:
  (a) Compute $P_T = \sum_{i=1}^{n} P_i$ and $Q = \sum_{i=1}^{n} Q_i$.
  (b) Verify whether the equation $\hat{e}(\sigma, P) = \hat{e}(H_2(m), P_T) \hat{e}(Q, P_0)$ holds. If so, the verifier accepts $(m, \sigma)$; otherwise the verifier rejects it.

  **Correctness of the proposed CL-SSMS scheme**

  The received message-signature pair $(m, \sigma)$ is accepted by the verifier since the following holds:

  \[
  \hat{e}(\sigma, P) = \hat{e}\left(\sum_{i=1}^{n} \sigma_i, P\right) = e\left(\sum_{i=1}^{n} (x_iH_2(m) + D_i), P\right) = e\left(\sum_{i=1}^{n} x_iH_2(m), P\right) \cdot e\left(\sum_{i=1}^{n} D_i, P\right) \quad \text{[due to bilinearity]}
  \]

  \[
  = \hat{e}(H_2(m), P) \cdot \hat{e}\left(\sum_{i=1}^{n} D_i, P\right) \quad \text{[due to bilinearity]}
  \]

  \[
  = \hat{e}(H_2(m), \sum_{i=1}^{n} P_i) \cdot \hat{e}\left(\sum_{i=1}^{n} sQ_i, P\right) \cdot [\because D_i = sQ_i]
  \]

  \[
  = \hat{e}(H_2(m), \sum_{i=1}^{n} P_i) \cdot \hat{e}\left(\sum_{i=1}^{n} sQ_i, P\right) \cdot [\because P_T = \sum_{i=1}^{n} P_i, Q = \sum_{i=1}^{n} Q_i, P_0 = sP]
  \]

  This assures the correctness of the proposed CL-SSMS scheme.

3.2. Proposed CL-SBMS scheme

In the broadcast multisignature scheme, the message issuer broadcasts the message $m$ to the group members $A = \{A_1,$

![Figure 2](image-url)  
*The proposed CL-SBMS scheme.*
}\$A_2, \cdots, A_n\$, and upon receiving \$m\$, each signer \$A_i(1 \leq i \leq n)\$ generates his own signature \$\sigma_i\$ on the same message \$m\$ simulta-

eously and then sends it to the designated clerk \$A_0\$ (say). Now \$A_0\$ verifies the individual signatures \$\sigma_i(1 \leq i \leq n)\$ and generates
the final short multisignature \$\sigma\$ on behalf of the group. The proposed complete CL-SBMS scheme consists of the following seven algorithms: Setup, Set-Secret-Value, Partial-

Private-Key-Extract, Set-Private-Key, Set-Public-Key, CL-

SBMS-Sign and CL-SBMS-Verify, where all these algorithms except CL-SBMS-Sign and CL-SBMS-Verify are the same and already discussed in the proposed CL-SSMS scheme. Thus, only CL-SBMS-Sign and CL-SBMS-Verify algorithms are discussed now. As an illustration, the proposed CL-SBMS is further described by the block diagram in Fig. 2.

- **CL-SBMS-sign**: For a given message \$m \in \{0, 1\}^*\$, each signer \$A_i(1 \leq i \leq n)\$ performs the following:
  - Computes \$\sigma_i = x_iH_2(m) + D_i\$.
  - Sends the message-signature pair \$(m, \sigma_i)\$ to the designated clerk \$A_0\$.
  - The clerk \$A_0\$ verifies the message-signature pairs \$(m, \sigma_i)\$ by determining whether the equation \$\hat{e}(\sigma_i, P) = \hat{e}(H_2(m), P_i)\$ holds.
  - If each of the pair \$(m, \sigma_i)(1 \leq i \leq n)\$ is valid, \$A_0\$ then computes the multisignature \$\sigma = \sum_i \sigma_i\$ and sends the final message-signature pair \$(m, \sigma)\$ to the verifier for verification.

- **CL-SBMS-verify**: In order to verify \$(m, \sigma)\$, the verifier carry out the following steps:
  - (a) Compute \$P_T = \sum_i P_i\$ and \$Q = \sum_i Q_i\$.
  - (b) Verify whether the equation \$\hat{e}(\sigma, P) = \hat{e}(H_2(m), P_T)\$ holds. If so, the verifier accepts \$(m, \sigma)\$; otherwise the verifier rejects it.

- **Correctness of the proposed CL-SBMS scheme**
  - (a) For the correctness of the proposed scheme, let us first check the individual message-signature pairs \$(m, \sigma_i)(1 \leq i \leq n)\$. Since \$P_0 = sP\$ and \$P_i = x_iP\$, \$D_i = sQ_i\$, \$\sigma_i = x_iH_2(m) + D_i\$. For \$1 \leq i \leq n\$, we have

\[
\hat{e}(\sigma_i, P) = \hat{e}(x_iH_2(m) + D_i, P) \\
= \hat{e}(x_iH_2(m), P)\hat{e}(D_i, P)[due\ bilinearity] \\
= \hat{e}(H_2(m), P)\hat{e}(x_i, P)[due\ bilinearity] \\
= \hat{e}(H_2(m), P)\hat{e}(Q_i, P_i)[due\ bilinearity] \\
= \hat{e}(H_2(m), P)\hat{e}(Q, P_0)[due\ bilinearity] \\
= \hat{e}(H_2(m), P)\hat{e}(Q, P_0)[due\ bilinearity]
\]

Thus, the message-signature pairs \$(m, \sigma_i)(1 \leq i \leq n)\$ is valid.

  - (a) The checking of the correctness of the final multisignature \$(m, \sigma)\$ as the last step is also valid as already proven in the earlier section.

4. Analysis of the proposed CL-SSMS and CL-SBMS schemes

The security and efficiency analyses of the proposed CL-SSMS and CL-SBMS schemes are discussed in this section. First, we analyze different security aspects of the two proposed short multisignature schemes. Then their efficiencies in terms of computation as well as communication costs are estimated.

4.1. Security analysis

It is known that the unforgeability against different types of adversaries is one of the most important security properties of any digital signature scheme, where unforgeability means only the group members are able to compute the valid multisignature on behalf of the group and no outsider(s) or a colluding subset of the group members can generate any of the proposed multisignature schemes. Based on the CL-PKC system (Al-Riyami and Paterson, 2003; Huang et al., 2006, 2007; Chen et al., 2008; Du and Wen, 2009; Choi et al., 2011), the unforgeability of any signature scheme involves two types of adversaries called Type I and Type II. The Type I adversary \$A_I\$ represents an outsider attacker who is able to replace the public key of any user with a value of his own choice, but he is unable to access the PKG’s master private key. This attack caused by the adversary \$A_I\$ is known as public key replacement attack (Gorlant and Saxena, 2005; Huang et al., 2006; Gangisetti et al., 2006; Huang et al., 2007; Chu and Zhao, 2008; Le and Gabillon, 2009; Biao et al., 2010; Islam and Biswas, 2012b; Islam and Biswas, 2013a). On the other hand, the Type II adversary \$A_{II}\$ acts as a malicious PKG (insider attacker) who is not allowed to replace users’ public keys, but can access the PKG’s master private key. This type of attack is called malicious PKG attack (Gorlant and Saxena, 2005; Huang et al., 2006; Gangisetti et al., 2006; Huang et al., 2007; Chu and Zhao, 2008; Le and Gabillon, 2009; Biao et al., 2010; Islam and Biswas, 2012b; Islam and Biswas, 2013a). Now, the security analysis of the proposed short multisignature schemes against different adversaries is given.

4.1.1. Unforgeability against the adversary \$A_I\$ (public key replacement attack resilience)

The adversary \$A_I\$ can replace the public key \$P_i\$ of the user \$A_i(1 \leq i \leq n)\$ with a value \$P_0\$ of his choice; however, he cannot access the master private key \$msk = s\$ of the PKG (Al-Riyami and Paterson, 2003; Huang et al., 2006; Huang et al., 2007; Chen et al., 2008; Du and Wen, 2009; Choi et al., 2011; Islam and Biswas, 2012b; Islam and Biswas, 2013a). Although \$A_I\$ knows the secret value \$x_i\$ corresponding to the replaced public key \$P_0(1 \leq i \leq n)\$, \$A_I\$ cannot forge any of the CL-SSMS and CL-SBMS schemes without the knowledge of the partial private key \$D_i\$ of the user \$A_i\$. Note that the partial private \$D_i = sQ_i(1 \leq i \leq n)\$ must be used to generate the individual signature \$\sigma_i = x_iH_2(m) + D_i\$, and it can be computed only if PKG’s master private key \$msk = s\$ is known. However, although \$s\$ can be extracted from \$P_0 = sP\$, the ECDLP is not solvable by any polynomial time-bounded algorithm (Koblitz, 1989; Silverman and Suzuki, 1998; Menezes et al., 1993; Frey et al., 1999; Gaudry, 2000), because it is known that such an algorithm does not exist. Thus, \$A_I\$ cannot generate \$\sigma_i = x_iH_2(m) + D_i\$ and the final signature \$\sigma\$ as well. Hence, the proposed CL-SSMS and CL-SBMS schemes are secure against the adversary \$A_I\$.

4.1.2. Unforgeability against the adversary \$A_{II}\$ (Malicious PKG attack resilience)

The adversary \$A_{II}\$ can access PKG’s master private key \$msk = s\$ but, the public key \$P_i\$ of the user \$A_i(1 \leq i \leq n)\$ is not allowed to be replaced by a value chosen by him.
(Al-Riyami and Paterson, 2003; Huang et al., 2006, 2007; Chen et al., 2008; Du and Wen, 2009; Choi et al., 2011; Islam and Biswas, 2012b; Islam and Biswas, 2013a). Since the master private key $msk = s$ is known to $\mathcal{A}_H$, he also knows the partial private key $D_i(1 \leq i \leq n)$ of the user $A_i(1 \leq i \leq n)$. However, the generation of the individual signature $s_i = x_iH_2(m) + D_i$ for $(1 \leq i \leq n)$ is only possible if the secret value $x_i(1 \leq i \leq n)$ is known to $\mathcal{A}_H$. Although he may try to derive $x_i$ from $P_i = x_iP(1 \leq i \leq n)$, he needs to solve the ECDLP in the elliptic curve group, which is not solvable in polynomial time. Thus, we can conclude that the forgery of the proposed CL-SSMS and CL-SBMS schemes is impossible by the adversary $\mathcal{A}_H$.

4.1.3. Unforgeability against normal adversary

Assume that an adversary $\mathcal{A}_M$ has knowledge about system’s parameter $\Omega = \{G, G_m, \hat{e}, q, P, P_0, H_1, H_2\}$ only and he tries to impersonate a user $A_i(1 \leq i \leq n)$. The adversary $\mathcal{A}_M$ can impersonate a user $A_i(1 \leq i \leq n)$ i.e., he can generate a forged multisignature on behalf of $A_i(1 \leq i \leq n)$, if PKG’s master private key $msk = s$ is known. If $s$ is disclosed to $\mathcal{A}_M$, then he can impersonate $A_i$ just by selecting a number $x_i \in \mathbb{Z}_q^*$ as his secret value since no public key certificate corresponding to the public key $P_i = x_iP$ is used in CL-PKC. Therefore, the adversary $\mathcal{A}_M$ can try to compute $msk = s$ from PKG’s public key $P_0 = sP$. However, due to the difficulties of solving the ECDLP in the elliptic curve group, the master private key $s$ cannot be extracted from $P_0$. So we can conclude that both the proposed multisignature schemes are secure against the adversary $\mathcal{A}_M$ under the ECDLP problem.

4.1.4. Achieving Girault’s trust level

In 1992, (Girault, 1992) defines different types of trust levels as given below, which must be achieved in designing an efficient digital signature scheme:

- **Level 1**: The PKG does not know the private keys but, it can still impersonate any user by generating false public keys that may be used without being detected.
- **Level 2**: The PKG can impersonate any user without being detected since PKG knows the users’ private keys.
- **Level 3**: The PKG cannot compute the private keys and if it generates false certificates for users, it can be detected.

For Trust Level 1, an adversary is allowed to replace the public key $P_i(1 \leq i \leq n)$ of $A_i(1 \leq i \leq n)$ by a false public key $P_i^*(1 \leq i \leq n)$ of his choice and thus, he knows the secret value $x_i(1 \leq i \leq n)$ corresponding to $P_i^*(1 \leq i \leq n)$. However, the master secret $msk = s$ is unknown to him. This trust level is equivalent to the adversary $\mathcal{A}_I$, and since it is discussed elaborately in Section 4.1.1, we can say that our schemes achieve Trust Level 1. Again from the definition of Trust Level 2, an adversary cannot replace the public keys $P_i(1 \leq i \leq n)$, but can compute the secret key $D_i = sQ_i(1 \leq i \leq n)$ of $A_i(1 \leq i \leq n)$ since he can access the master secret $msk = s$ of PKG. Hence, the Trust Level 2 is actually $\mathcal{A}_H$ and our scheme can achieve this trust level also as discussed in Section 4.1.2. According to the definition of Trust Level 3 (Girault, 1992; Gorantla and Saxena, 2005), an untrusted PKG, who can replace the public keys $P_i = x_iP(1 \leq i \leq n)$ with false public keys $P_i^* = x_i^*P(1 \leq i \leq n)$ of his choice, does not have the knowledge about the secret key $x_i(1 \leq i \leq n)$, but can still impersonate a user $A_i(1 \leq i \leq n)$ without being detected, since the partial private keys $D_i = sQ_i(1 \leq i \leq n)$ are known to PKG. Thus, the untrusted PKG can compute a forged multisignature by generating another valid key pair $(D_i, P_i^*)$ for $(1 \leq i \leq n)$ corresponding to the user $A_i(1 \leq i \leq n)$. Here we have shown that the proposed signature schemes also achieve Girault’s Trust Level 3. In our schemes, user $A_i$ generates his secret value $x_i$, and the corresponding public key $P_i = x_iP$, and generates the partial private keys as $D_i = sQ_i$, where $Q_i = H_1(ID_i, P_i)$. Therefore, user $A_i$ owns only one partial private key $D_i$ corresponding to the public key $P_i = x_iP$ and he cannot generate another false public key $P_i^* = x_i^*P$ by maintaining the same partial private key $D_i$. However, PKG can compute another pair $(D_i, P_i^*) (1 \leq i \leq n)$ on behalf of $A_i(1 \leq i \leq n)$ and it can be detected easily because only he has that ability.

4.2. Performance analysis

This section analyzes the performance of the proposed CL-SSMS and CL-SBMS schemes, where the computation cost and communication cost (signature length) are considered. For this, we use the method adopted in (Barreto et al., 2004; Chung et al., 2007; Ren et al., 2007; Tan et al., 2010; Cao et al., 2010; He et al., 2011; Islam and Biswas, 2012a), where the bilinear pairing (Tate pairing) is defined over the supersingular elliptic curve $E/F_p$: $y^2 = x^3 + x$ with embedding degree 2 to achieve 1024-bit RSA level security, and the Solinas prime $q = 2^{159} + 2^{17} + 1$ is a 160-bit number (Solinas, 2011), and $p$ is a 512-bit prime satisfying $p + 1 = 12q$. In addition, we consider the running time calculated for different cryptographic operations in (Ren et al., 2007; Cao et al., 2010; He et al., 2011) using MIRACAL software (Shamus Software Ltd, 1988) and implemented on a Pentium IV 3 GHZ processor with 512 MB RAM and the Windows XP (Microsoft) operating system. Furthermore, Chung et al. (2007) indicate that the time needed to execute the modular exponentiation $(T_{MA})$ is approximately $240T_{MA}$, where $T_{MA}$ represents the time complexity of executing the modular multiplication. It was also

<table>
<thead>
<tr>
<th>Table 1</th>
<th>Notations and descriptions of various cryptographic operations and their operational time (in milliseconds).</th>
</tr>
</thead>
<tbody>
<tr>
<td>Notations</td>
<td>Descriptions</td>
</tr>
<tr>
<td>$T_{EM}$</td>
<td>Time complexity for executing the elliptic curve scalar point multiplication, $T_{EM} \approx 6.38$ ms</td>
</tr>
<tr>
<td>$T_{BP}$</td>
<td>Time complexity for executing the bilinear pairing operation, $T_{BP} \approx 20.01$ ms</td>
</tr>
<tr>
<td>$T_{PX}$</td>
<td>Time complexity for executing pairing-based exponentiation, $T_{PX} \approx 11.20$ ms</td>
</tr>
<tr>
<td>$T_{EX}$</td>
<td>Time complexity for executing the modular exponentiation, $T_{EX} \approx 55.20$ ms</td>
</tr>
<tr>
<td>$T_{EA}$</td>
<td>Time complexity for executing the addition of two elliptic curve points, which is negligible</td>
</tr>
</tbody>
</table>
mentioned in (Cao et al., 2010; He et al., 2011) that the time needed to execute one bilinear pairing (Tate pairing) operation ($T_{BP}$) is approximately 20.01 ms i.e., $T_{BP} \approx 20.01$ ms, and from the works proposed in (Barreto et al., 2004; Tan et al., 2010), we obtained $T_{BP} \approx 3T_{EM} \approx 87T_{ML}$, where $T_{EM}$ indicates the time complexity for executing one elliptic curve scalar point multiplication. Thus, the execution of one modular exponentiation ($T_{EX}$) operation on a Pentium IV 3 GHZ processor with 512 MB RAM and Windows XP takes about $(20.01 \times 240) / 87 \approx 55.2$ ms i.e., $T_{EX} \approx 55.2$ ms. The definition, description and the running time (in milliseconds) of various cryptographic operations are presented in Table 1.

Now, we compare the two proposed multisignature schemes with the multisignature schemes available in the literature (Gangishetti et al., 2006; Giri and Srivastava, 2007; Chu and Zhao 2008; Le and Gabillon, 2009; Biao et al., 2010; Yang et al., 2010; Gui and Zhang, 2010). Table 2 summarizes the same in terms of the notations given in Table 1, whereas the corresponding numerical values (running time) in terms of $n$ (number of signers) are given in Table 3. It can be seen that two existing multisignature schemes (Gangishetti et al., 2006; Chu and Zhao, 2008) require comparatively lower computation costs and are $(76.72n - 6.38) ms$ and $(55.2n + 110.4) ms$, respectively, whereas our proposed methods have reported the lowest computation costs of $46.4n$ ms for both of the multisignature schemes. Similarly, the communication cost of our proposed schemes as shown in Table 2 has the lowest cost by using single elliptic curve point.

Furthermore, a multisignature scheme is called computation and communication efficient if the following two requirements are achieved:

- **Fixed length signature**: A multisignature scheme is called fixed length if the length of the final multisignature is the same as the length of the individual signature generated by each signer. In our schemes (CL-SSMS and CL-SBMS), the lengths of the both individual signature $\sigma_i$ for ($1 \leq i \leq n$) and the final multisignature $\sigma$ are equal to an elliptic curve point (1P). So the proposed CL-SSMS and CL-SBMS are the fixed length multisignature schemes.

- **Constant verification time**: A multisignature has constant verification time if the time needed to verify either the final multisignature or an individual signature is the same. In case of the proposed CL-SSMS scheme, the verification equation $\hat{e}(\sum_{i=1}^{m}P_i, \sum_{i=1}^{m}Q_i) \neq \hat{e}(H_2(m), \sum_{i=1}^{m}P_i)$ shows that only two bilinear pairing operations are executed to verify an individual signature $\sigma_i$ and where the other computations such as $\sum_{i=1}^{m}P_i, \sum_{i=1}^{m}Q_i$, and $\hat{e}(\sum_{i=1}^{m}Q_i, P_0)$ can be computed offline since $P_0$, $P_i$, and $Q_i (1 \leq i \leq n)$ are known publicly. Thus, the time needed to verify an individual signature of a signer is $2T_{BP} \approx 40.02$ ms. On the other hand, the final multisignature ($m, \sigma$) can be verified by using the equation $\hat{e}(\sigma, P) = \hat{e}(H_2(m), \sum_{i=1}^{m}P_i) \neq \hat{e}(Q, P_0)$ and accordingly the verification time is also $2T_{BP} \approx 40.02$ ms. Hence, the proposed CL-SSMS scheme achieves the constant verification time attribute. Similarly, the other proposed CL-SBMS scheme also satisfies a constant verification time.

In Table 4, we compare our schemes in terms of cryptosystems used by various existing schemes (Gangishetti et al., 2006; Giri and Srivastava, 2007; Chu and Zhao 2008; Le and Gabillon, 2009; Biao et al., 2010; Yang et al., 2010; Gui and Zhang, 2010). Table 2 summarizes the computation cost of our proposed schemes as shown in Table 2 has the lowest cost by using single elliptic curve point.

<p>| Table 2: Comparison of the proposed CL-SSMS and CL-SBMS schemes with others. |
|---------------------------------|------------------|------------------|------------------|</p>
<table>
<thead>
<tr>
<th>Schemes</th>
<th>Signature length</th>
<th>Signature generation cost</th>
<th>Overall computation cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gangishetti et al (2006)</td>
<td>Fixed length</td>
<td>$2mP + nT_{EX} + mT_{ML} + nT_{ML}$</td>
<td>$55.2 m + 110.4$</td>
</tr>
<tr>
<td>Giri and Srivastava (2007)</td>
<td>Fixed length</td>
<td>$2mP + nT_{EX} + mT_{ML} + nT_{ML}$</td>
<td>$55.2 m + 110.4$</td>
</tr>
<tr>
<td>Chu and Zhao (2008)</td>
<td>Fixed length</td>
<td>$2mP + nT_{EX} + mT_{ML} + nT_{ML}$</td>
<td>$55.2 m + 110.4$</td>
</tr>
<tr>
<td>Le and Gabillon (2009)</td>
<td>Fixed length</td>
<td>$2mP + nT_{EX} + mT_{ML} + nT_{ML}$</td>
<td>$55.2 m + 110.4$</td>
</tr>
<tr>
<td>Biao et al. (2010)</td>
<td>Fixed length</td>
<td>$2mP + nT_{EX} + mT_{ML} + nT_{ML}$</td>
<td>$55.2 m + 110.4$</td>
</tr>
<tr>
<td>Yang et al. (2010)</td>
<td>Fixed length</td>
<td>$2mP + nT_{EX} + mT_{ML} + nT_{ML}$</td>
<td>$55.2 m + 110.4$</td>
</tr>
<tr>
<td>Proposed CL-SSMS</td>
<td>Fixed length</td>
<td>$2mP + nT_{EX} + mT_{ML} + nT_{ML}$</td>
<td>$55.2 m + 110.4$</td>
</tr>
<tr>
<td>Proposed CL-SBMS</td>
<td>Fixed length</td>
<td>$2mP + nT_{EX} + mT_{ML} + nT_{ML}$</td>
<td>$55.2 m + 110.4$</td>
</tr>
</tbody>
</table>
The schemes of (Giri and Srivastava, 2007; Chu and Zhao, 2008; Le and Gabillon, 2009) are designed based upon PKI and thus suffer from the overhead of public key certificate storage and management, whereas the schemes (Gangishetti et al., 2006; Biao et al., 2010; Yang et al., 2010; Gui and Zhang, 2010) have private key escrow problem as they are developed based upon IBC. Since our CL-SSMS and CL-SBMS schemes are designed based upon CL-PKC, which is superior to both PKI and IBC cryptosystems, the proposed multisignature methods are also more efficient than (Gangishetti et al., 2006; Giri and Srivastava, 2007; Chu and Zhao, 2008; Le and Gabillon, 2009; Biao et al., 2010; Yang et al., 2010; Gui and Zhang, 2010).

5. Conclusions

In this paper, two certificateless short multisignature schemes CL-SSMS and CL-SBMS using elliptic curve and bilinear pairing are proposed, where the former scheme is suitable for sequential architectures and the latter one is suitable for parallel architectures. The proposed schemes have been developed using the most efficient CL-PKC cryptosystems, which are free from the public key certificate management burden and the private key escrow problem. The security analysis has been provided and shown that the proposed schemes are secure against both Type I and Type II adversaries existent in any CL-PKC cryptosystem. Moreover, both the schemes produce short signatures of length equal to a single elliptic curve point and require constant verification time. Thus they are applicable in resource-constrained environments where communication bandwidth, battery life, computing power, and storage space are limited. However, the proposed CL-SSMS and CL-SBMS schemes suffer from the execution of costly elliptic curve bilinear pairing and Map-To-Point hash functions. In addition, a super-singular elliptic curve group with a large group-size is required for realization of bilinear pairing, and Map-To-Point hash function implementation is a probabilistic approach. Therefore, the short multisignature scheme without bilinear pairing and Map-To-Point hash function would be attempted for further improvement and suitable for real-life applications.

Acknowledgements

The authors are grateful to the Editor-in-Chief, Prof. Mansour M. Alsulaiman and anonymous reviewers for their valuable comments and suggestions that helped to improve this paper. This research work is supported by the Department of Science and Technology (DST), Govt. of India under the INSPIRE fellowship Ph.D. program (Grant No. IF10247) and the Department of Information Technology (DIT), Ministry of Communication and Information Technology, Govt. of India under the Information Security Education and Awareness (ISEA) program (Project No. MIT(2)/2006-08/189/CSE). The authors would also like to express their gratitude and heartiest thanks to the Department of Computer Science and Engineering, Indian School of Mines, Dhanbad-826004, India for providing their research support, as without such help this work could not have been carried out.

References

Biao, W., Xiaodong, Y., Guang, Y., 2010. An Identity-Based Multisignature Scheme from the Weil Pairing. In: Proceedings of...


