Forward-Secure Multisignature, Threshold Signature and Blind Signature Schemes

Jia Yu
1College of Information Engineering, Qingdao University, Qingdao, P. R. China

Fanyu Kong
2Institute of Network Security, Shandong University, Jinan, P. R. China

Xiangguo Cheng, Rong Hao, Yangkui Chen, Xuliang Li
1College of Information Engineering, Qingdao University, Qingdao, P. R. China

Guowen Li
3School of Computer Science and Technology, Shandong Jianzhu University, Jinan, P. R. China

Abstract—Forward-secure signatures are proposed to tackle the key exposure problem, in which the security of all signatures prior to key leakage is still kept even if the secret key leaks. In this paper, we construct two forward-secure multisignature schemes, one forward-secure threshold signature scheme, and one forward-secure blind signature scheme. Our constructions are based on the recently proposed forward-secure signature scheme from bilinear maps in [11]. Our constructions are very efficient and useful thanks to the elegant structure of the base scheme. Such schemes play an important role in many electronic applications such as cryptographic election systems, digital cash schemes, and e-cheques.

Index Terms—multisignature; threshold signature; secret sharing; blind signature; forward security

I. INTRODUCTION

Forward security for digital signature is proposed to deal with the key exposure problem. In a forward-secure signature scheme, the whole time is divided into discrete time periods. Different secret keys are used to sign the messages in different time periods, while the public key is unchanged during the whole lifetime. The new secret key for the next time period is computed from the old one by a one-way key update paradigm. Each signature is associated with one time period. When the signature is verified, we also need to verify the consistency of the time period. Exposure of the current secret key does not help the adversary to forge a valid signature of previous time period in this primitive.

Forward-secure signature was firstly proposed by Anderson [1], and then formalized by Bellare and Miner [2]. Bellare and Miner also gave the definition of forward-secure signature scheme and its security. Subsequently some constructions of forward-secure signature schemes [3–6] were proposed, which had different trade-offs among key size, signing time and update time. The scheme [5] had optimal signing and verifying algorithms at the expense of slower key update. In comparison, the scheme [6] could achieve fast key update but had slower signing and verifying algorithms. Malkin et al. [7] proposed generic forward-secure signatures with an unbounded number of time periods. Hierarchical ID-based cryptography could be used to construct forward-secure signature schemes. Based on the hierarchical ID-based cryptography [8], some forward-secure signature scheme using bilinear maps were proposed in [9–11]. Boyen et al. presented a forward-secure signature with untrusted update [12], in which the secret key is additionally protected by an extra secret that is possibly derived from a password and key update procedure can be completed by the encrypted version of signing key. Libert et al. [13] gave generic constructions of forward-secure signatures in untrusted update environments.

Forward-secure symmetric-key encryption was studied in [14] and forward-secure public key encryption was also studied in [15]. Forward-secure threshold signatures were researched in [16–19]. Key-insulation [20–23] and intrusion-resilient cryptography [24–27] can achieve a higher level of security than forward-secure cryptography. However, these methods were not able to apply to many scenarios.

Multisignature was firstly proposed by Itakura and Nakamura [28]. Multisignature allows any subgroup of users to cooperate to sign a message. The verifier can assure that any user participates in signing. The security of multisignature was formalized in [29]. Forward-secure multisignature was studied in [30].

Threshold signature is one kind of distributive signatures. In a $(t+1, n)$ threshold signature, a secret key is distributed into $n$ users, and each user has one share of the secret key. Only more than $t$ users can jointly generate signatures by a reconstruction procedure. The first forward-secure threshold signature was proposed by
Abdalla et al. [16]. However, in their scheme the size of both the public key and the secret key are very large, what’s more, the scheme needs a lot of interactions. Following by Abdalla’s work, another forward-secure threshold signature with proactive property [17] is proposed, which needs shorter keys. Paper [19] proposed an efficient forward-secure threshold signature scheme from bilinear maps.

Blind signature introduced by David Chaum [31], is a form of digital signature in which the content of a message is blinded before it is signed. The generating blind signature can be verified against the original, unblinded message in the manner of a standard digital signature. Blind signature plays an important role in cryptographic election systems and digital cash schemes. Recently, some papers about blind signatures were proposed in [30, 35]. Forward-secure blind signatures were proposed in [32-34].

Our contribution. We construct two forward-secure multisignature schemes, one forward-secure threshold signature scheme, and one forward-secure blind signature scheme based on the recently proposed forward-secure signature scheme from bilinear maps [11]. Our constructions very efficient and useful thanks to the base scheme. Such schemes are very important for many e-commerce applications. Our schemes are forward-secure in random oracle model assume CDH problem is hard.

II. PRELIMINARIES

A. Cryptographic Assumption

We review some cryptographic preliminaries which have been introduced in many papers.

Let $G_1$ and $G_2$ be two cyclic groups of prime order $q$, where $G_1$ and $G_2$ are represented additively and multiplicatively, respectively. $P \in G_1$ is a generator of $G_1$. A bilinear map $\hat{e}: G_1 \times G_1 \rightarrow G_2$ satisfies:

1. Bilinear: For all $P, Q \in G_1$ and $a, b \in \mathbb{Z}$, there is $\hat{e}(aP, bQ) = (\hat{e}(P, Q))^a$.

2. Non-degenerate: The map does not send all pairs in $G_1 \times G_1$ to the identity in $G_2$.

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

Computation Diffie-Hellman (CDH) problem: Given $(P, aP, bP)$, where $a, b \in \mathbb{Z}_q$, compute $abP$.

Definition 1 (CDH Assumption). A probabilistic algorithm $A$ is said $(t, \varepsilon)$-attack CDH problem in $G_1$ if $A$ runs at most time $t$, computes CDH problem with an advantage of at least $\varepsilon$. We say that $G_1$ is a $(t, \varepsilon)$-secure CDH group if no probabilistic algorithm $A$ $(t, \varepsilon)$-attack CDH problem in $G_1$.

B. Forward-Secure Multisignature Scheme

A forward-secure signature scheme consists of a key generation algorithm, a key update algorithm, a signing algorithm and a verifying algorithm.

Definition 2 (Forward-secure Multisignature Scheme). A forward-secure multisignature scheme is a quadruple of algorithms $\text{FMSIG}=(\text{FMSIG.key}, \text{FMSIG.update}, \text{FMSIG.sign}, \text{FMSIG.verify})$, where:

1. $\text{FMSIG.key}$: the key generation algorithm, is a probabilistic algorithm which takes as input a security parameter $k \in \mathbb{N}$ and the total number of time periods $T$, and generates a public key $PK$ and the initial secret key $SK_{0}^{(i)}$ for signer $j(i=1, \ldots, n)$.

2. $\text{FMSIG.update}$: the key update algorithm, is a probabilistic algorithm which takes as input the secret key $SK_{j}^{(i)}$ signer $j$ holds of the current period $i$ and generates the new secret key $SK_{i+1}^{(j)}$ for the next period.

3. $\text{FMSIG.sign}$: the signing algorithm, takes as input the secret key $SK_{j}^{(i)}$ signer $j$ holds of the current time period $i$ and a message $M$, and generates a partial signature. All partial signature can generate the final multisignature $<i, \text{sign}>$ of $M$ for period $i$. This algorithm may be probabilistic.

4. $\text{FMSIG.verify}$: the verifying algorithm, is a deterministic algorithm which takes as input the public key $PK$ , a message $M$ and a candidate signature $<i, \text{sign}>$, and output 1 when $<i, \text{sign}>$ is a valid signature or 0, otherwise.

C. Forward-Secure Threshold Signature Scheme

A forward-secure threshold signature scheme consists of a key generation algorithm, a key update algorithm, a signing algorithm and a verifying algorithm.

Definition 3 (Forward-secure Threshold Signature Scheme). A forward-secure threshold signature scheme is a quadruple of algorithms $\text{FTSIG}=(\text{FTSIG.key}, \text{FTSIG.update}, \text{FTSIG.sign}, \text{FTSIG.verify})$, where:

1. $\text{FTSIG.key}$: the key generation algorithm, is a probabilistic algorithm which takes as input a security parameter $k \in \mathbb{N}$ and the total number of time periods $T$, and generates a public key $PK$ and the initial secret key $SK_{0}^{(i)}$ for player $j(i=1, \ldots, n)$.

2. $\text{FTSIG.update}$: the key update algorithm, is a probabilistic algorithm which takes as input all the secret key $SK_{j}^{(i)}$ players $j(i=1, \ldots, n)$ hold of the current period $i$ and generates the new secret key $SK_{i+1}^{(j)}$ for the next period.

3. $\text{FTSIG.sign}$: the signing algorithm, takes as input all the secret key $SK_{j}^{(i)}$ players $j(i=1, \ldots, n)$ hold of the
current time period $i$ and a message $M$, and any $t$ players generate the final threshold signature $<i, \text{sign}>$ of $M$ for period $i$. This algorithm may be probabilistic.

4. **FMSIG.verify**: the verifying algorithm, is a deterministic algorithm which takes as input the public key $PK$, a message $M$ and a candidate signature $<i, \text{sign}>$, and output 1 when $<i, \text{sign}>$ is a valid signature or 0, otherwise.

### D. Forward-Secure Blind Signature Scheme

A forward-secure blind signature scheme consists of a key generation algorithm, a key update algorithm, a signing algorithm and a verifying algorithm.

**Definition 4 (Forward-Secure Blind Signature Scheme).** A forward-secure blind signature scheme is a quadruple of algorithms $\text{FBSIG} = (\text{FBSIG.key}, \text{FBSIG.update}, \text{FBSIG.sign}, \text{FBSIG.verify})$, where:

1. **FBSIG.key**: the key generation algorithm, is a probabilistic algorithm which takes as input a security parameter $k \in \mathbb{N}$ and the total number of time periods $T$, and generates a public key $PK$ and the initial secret key $SK_0$.

2. **FBSIG.update**: the key update algorithm, is a probabilistic algorithm which takes as input the secret key $SK_i$ of the current period $i$, and generates the new secret key $SK_{i+1}$ for the next period.

3. **FBSIG.sign**: the signing algorithm, takes as input the secret key $SK_j$ of the current time period $j$ and a message $M$.
   - (1) **Blind**: On a random string $r$ and a message $M$ as the input, it outputs a string $R$ and sends it to the signer.
   - (2) **Sign**: On a string $R$ and the secret key as the input, it outputs a blind signature $<i, \text{sign}>$.
   - (3) **Unblind**: On a blind signature $<i, \text{sign}>$ and random string $r$ as the input, it outputs the final unblind signature $<i, \text{sign}>$.

4. **FMSIG.verify**: the verifying algorithm, is a deterministic algorithm which takes as input the public key $PK$, a message $M$ and a candidate signature $<i, \text{sign}>$, and output 1 when $<i, \text{sign}>$ is a valid signature or 0, otherwise.

### III. THE PROPOSED SCHEMES

#### A. Notations

Our schemes use a binary tree in [11], which is firstly suggested to form forward-secure signature schemes by Bellare and Miner [2]. The notations description is the same as the description in [11]. We omit the description of the notations here. Please refer to [11].

### B. Review the Forward-Secure Signature Scheme in [11]

We review the basis forward-secure signature scheme here. The description of this scheme is taken from [11] directly.

Let $IG$ be a CDH parameter generator, therefore the CDH assumption holds.

(1) algorithm $\text{FBSIG.key}(k, l, T)$

Begin

Run $IG(1^k)$ to generate additive group $G_1$ and multiplicative group $G_2$ with the same primitive order $q$ and an admissible pairing $\hat{e}: G_1 \times G_1 \rightarrow G_2$.

Select cryptographic hash functions $H_1 : \{0, 1\}^* \times G_1 \rightarrow \mathbb{Z}_q^*$, $H_2 : G_1 \rightarrow G_1$, and $H_3 : \{0, 1\}^* \times G_1 \rightarrow G_1$.

Select generator $P \in G_1$ and secret $s_r \in \mathbb{Z}_q^*$, and let $Q = s_r P$, $S_q = s_q H_2(Q)$.

Let the public key be $PK = \{G_1, G_2, \hat{e}, H_1, H_2, H_3, P, Q\}$.

Select $s_q, s_0 \in \mathbb{Z}_q^*$, and compute $Q_0 = s_0 P$, $Q_1 = s_1 P$.

Compute $S_0 = Q_0 + s_0 H_1(0, Q_0) H_2(Q)$ and $S_1 = S_0 + s_1 H_1(1, Q_0) H_2(Q)$.

For $j = 1$ to $l - 1$

- Select $s_{0,j}, s_{1,j} \in \mathbb{Z}_q^*$, and compute $Q_{0,j} = s_{0,j} P$, $Q_{1,j} = s_{1,j} P$.
- Compute $S_0 = S_0 + s_0 H_1(0, Q_{0,j}) H_2(Q)$, and $S_1 = S_0 + s_1 H_1(1, Q_{0,j}) H_2(Q)$.

End

(2) algorithm $\text{FBSIG.update}(SK_i)$

Begin

If $i = T - 1$ then

Let $SK_T = \phi$.

Else

- Parse $<i > \in \{0, 1\}^*$, $i, i_0 \in \{0, 1\}$
- $SK_i = \{S_{0,i}, S_{1,i}, M_{0,i}, M_{1,i}\}$
- And $M_{0,i} = Q_{0,0}, \ldots, Q_{0,i}, Q_{0,i+1}, \ldots, Q_{0,T}$
- If $i_0 = 0$ then

    - Find $<S_{0,i+1}, Q_{0,i+1}, M_{0,i+1}>$ from $Set_{<i>}$.

© 2010 ACADEMY PUBLISHER
and set $S_{i_{2...i_1}} = S_{i_{2...i_1}} - \{< S_{i_{3...i_1}}, Q_{i_{3...i_1}}> \}$.
Set $M_{i_{2...i_1}} = (M_{i_{3...i_1}} - (Q_{i_{3...i_1}})) \cup \{Q_{i_{3...i_1}}\}$.
Set $\text{SK}_{i_{1}} = (S_{i_{1}}, S_{i_{2...i_1}}, M_{i_{2...i_1}})$.
}
}
Else
Find the maximal $j (1 \leq j < l)$ satisfying $i_j = 0$,
and let $\eta = [i_1 \ldots i_{j-1}, i_0 = \varepsilon]$.
Here $<i+1 > = \eta^{0^{\ldotsj}}$.
Find $< S_{\eta}, Q_{\eta} >$ from $S_{\eta}$,
Set $M_{i_{2...i_1}} = M_{i_{3...i_1}} \cup \{Q_{\eta}\}$,
and set $S_{i_{2...i_1}} = S_{i_{3...i_1}} - \{< S_{\eta}, Q_{\eta} > \}$.
Set $M_{i_{2...i_1}} = M_{i_{3...i_1}} - \{Q_{\eta_{i_{3...i_1}}}, Q_{\eta_{i_{3...i_2}}}, \ldots, Q_{\eta_{i_{4...i_3}}} >$.
}

For $m = 1$ to $l - j$
{
Select $s^{\eta}, s^{\eta_{i_{1}}} \in \mathbb{Z}_q$,
and compute $Q^{\eta} = s^{\eta} P$,
$Q_{\eta} = s^{\eta_{i_{1}}} P$,
$S^{\eta} = S^{\eta_{i_{1}}} + s^{\eta} H_{2}(q^{\eta_{i_{1}}} Q^{\eta} - H_{2}(Q))$,
and $S^{\eta_{i_{1}}} = S^{\eta_{i_{1}}} + s^{\eta_{i_{1}}} H_{2}(q^{\eta_{i_{1}}} - 1) . Q_{\eta_{i_{1}}} - H_{2}(Q)$.
Set $S_{i_{1}} = S_{i_{2...i_1}} \cup \{s^{\eta_{i_{1}}} Q^{\eta_{i_{1}}} > \}$,
and $M_{i_{1}} = M_{i_{2...i_1}} - \{Q^{\eta} >$.
}
Set $\text{SK}_{i_{1}} = (S_{i_{1}}, S_{i_{2...i_1}}, M_{i_{2...i_1}})$.
}
Erase all interim data, and return $\text{SK}_{i_{1}}$.
End

(3) algorithm $\text{FSIG}.\text{sign}(i, \text{SK}_{i}, M)$
Begin
Parse $< i := i_{2...i_1} \text{ }, \text{SK}_{i} = \{S_{i_{2...i_1}}, M_{i_{2...i_1}}\}$.
Select $r \in \mathbb{Z}_q$, and compute $U = rP$.
Compute $V = S_{i_{2...i_1}} + r H_{2}(i_{2...i_1} || M, U)$.
Return signature $<i, \text{sign} = (U, V, M_{i_{2...i_1}})$.
End

(4) algorithm $\text{FSIG}.\text{verify}(M, PK, < i, \text{sign} >)$
Begin
Parse $< i := i_{2...i_1} \text{ }, \text{sign} = (U, V, M_{i_{2...i_1}})$,
and $M_{i_{2...i_1}} = (Q_{i_{2...i_1}}, \ldots, Q_{i_{1}})$.
Verify:
\begin{equation}
\tilde{e}(P, V) = \tilde{e}(Q + \sum_{j=1}^{l} H_{2}(i_{2...i_1} \text{ } Q_{i_{3...i_1}}, Q_{i_{3...i_2}}, \ldots, Q_{i_{4...i_3}})) \cdot \hat{e}(U, H_{2}(i_{2...i_1} \text{ } || M, U))
\end{equation}
If it holds, return "valid", otherwise, return "invalid".
End

C. The Proposed Forward-Secure Multisignature Schemes
We give two forward-secure multisignature schemes in this subsection. The first scheme has robust property but needs more computations. The second has not robust property but needs fewer computations than the first scheme.

Scheme 1:
Each signer uses the same key algorithm and update algorithm to generate the public key and the secret key in time $i$. For convenience, we denote the secret key signer $j$ holds in time period $i$ $\text{SK}^{(j)} = \{S^{(j)}, \text{Set}^{(j)}(j), M^{(j)}\}$, where $M^{(j)} = (Q^{(j)}, \ldots, Q^{(j)}_{i_{2...i_1}})$.

We describe the signing algorithm and the verifying algorithm as follows.

$\text{FMSIG}.\text{sign}(i, \text{SK}_{i}, M)$
Begin
Parse $< i := i_{2...i_1} \text{ }, \text{SK}_{i} = \{S_{i_{2...i_1}}, \text{Set}_{i_{2...i_1}}, M_{i_{2...i_1}}\}$.
Select $r^{(i)} \in \mathbb{Z}_q$, and compute $U^{(i)} = r^{(i)} P$.
Compute $V^{(i)} = S^{(i)} + r^{(i)} H_{2}(i_{2...i_1} || M, U^{(i)})$.
The partial signature is $< i, \text{sign} = (U^{(i)}, V^{(i)}, M^{(j)}) >$.
We can use the following equation to verify whether the partial signature is valid or not:
\begin{equation}
\tilde{e}(P, V^{(j)}) = \tilde{e}(Q^{(j)} + \sum_{j=1}^{l} H_{2}(i_{2...i_1} \text{ } Q^{(j)}_{i_{2...i_1}}, Q^{(j)}_{i_{3...i_2}}, \ldots, Q^{(j)}_{i_{4...i_3}}) \cdot \hat{e}(U^{(i)}, H_{2}(i_{2...i_1} || M, U^{(i)}))
\end{equation}
Compute $V = \sum_{j=1}^{*} V^{(j)}$.
Let $\xi^{(i)} = \{M^{(i)}_{i_{3...i_1}}, \ldots, M^{(i)}_{i_{2...i_1}}\}$.
The final signature is $< i, \text{sign} = (\{U^{(i)}\}_{j=1...n}, V, \xi^{(i)}) >$.
End

$\text{FMSIG}.\text{verify}(M, PK, < i, \text{sign} >)$
Begin
Parse $< i := i_{2...i_1} \text{ }, \text{sign} = (\{U^{(i)}\}_{j=1...n}, V, \xi^{(i)})$,
and $\xi^{(i)} = \{M^{(i)}_{i_{3...i_1}}, \ldots, M^{(i)}_{i_{2...i_1}}\}$,
where $M^{(i)}_{i_{2...i_1}} = (Q^{(i)}_{i_{2...i_1}}, \ldots, Q^{(i)}_{i_{3...i_2}}, \ldots, Q^{(i)}_{i_{4...i_3}})$.
Verify:
\begin{equation}
\tilde{e}(P, V) = \tilde{e}(Q + \sum_{j=1}^{l} H_{2}(i_{2...i_1} \text{ } Q^{(i)}_{i_{3...i_1}}, Q^{(i)}_{i_{3...i_2}}, \ldots, Q^{(i)}_{i_{4...i_3}})) \cdot \hat{e}(U^{(i)}, H_{2}(i_{2...i_1} || M, U^{(i)}))
\end{equation}
If it holds, return "valid", otherwise, return "invalid".
End
The public commits include \( q = \zeta, \sum_{i=1}^{n} H_i(i_{i_2},...,i_{d}) \). We modify the procedure as follows:

\[
\begin{align*}
\hat{e}(P, V) &= \hat{e}(\sum_{j=1}^{n} V^{(j)}), \\
&= \prod_{j=1}^{n} \hat{e}(P, V^{(j)}) \\
&= \prod_{j=1}^{n} \hat{e}(P, S_{i_j}^{(j)} + r^{(j)} H_i(i_{i_2},...,i_{d}) \parallel M, U^{(j)})) \\
&= \prod_{j=1}^{n} \hat{e}(P, S_{i_j}^{(j)} + r^{(j)} H_i(i_{i_2},...,i_{d}) \parallel M, U^{(j)})) \\
&= \prod_{j=1}^{n} \hat{e}(P, s^{(j)} H_2(Q^{(j)}) + \sum_{d=1}^{r} q^{(j)} H_i(i_{i_2},...,i_{d}) Q^{(j)}(i_{j})) \\
&= \prod_{j=1}^{n} \hat{e}(P, H_i(i_{i_2},...,i_{d}) \parallel M, U^{(j)})) \\
&= \prod_{j=1}^{n} \hat{e}(H_2(Q^{(j)}), s^{(j)} P + \sum_{d=1}^{r} q^{(j)} H_i(i_{i_2},...,i_{d}) Q^{(j)}(i_{j})) \\
&= \prod_{j=1}^{n} \hat{e}(H_2(Q^{(j)}), H_i(i_{i_2},...,i_{d}) \parallel M, U^{(j)})) \\
\end{align*}
\]

Scheme 2:

In order to make the verifying algorithm simpler, we construct another forward-secure multisignature scheme by modifying the key algorithm and update algorithm as follows:

(1) The first modification is in key algorithm. All signer \( j=1,\ldots,n \) compute \( Q^{(j)} = s^{(j)} P \) and \( Q = \sum_{j=1}^{n} Q^{(j)} \). The public key is \( PK = \{G_1, G_2, \hat{e}, H_i, H_2, H_3, P, Q\} \).

(2) The second modification is in key algorithm and update algorithm. We consider all the operations as the following stations:

Select \( s_\zeta \in \mathbb{Z}_q^* \), and compute \( Q_\zeta = s_\zeta P \), and \( S_{\zeta} = S_{\zeta, i_j} + s_\zeta H_i(\zeta, Q_\zeta) H_2(Q) \) for some \( \zeta(\zeta = d) \).

We modify the procedure as follows:

All signer \( j=1,\ldots,n \) select \( s^{(j)}_\zeta \in \mathbb{Z}_q^* \), and compute and broadcast \( Q^{(j)} = s^{(j)}_\zeta P \), and then compute \( Q_\zeta = \sum_{j=1}^{n} Q^{(j)} \) and \( S_{\zeta} = S_{\zeta, i_j} + s^{(j)}_\zeta H_i(\zeta, Q_\zeta) H_2(Q) \).

Set \( M_{\zeta, i_j} = M_{i_j} = (Q_\zeta, \ldots, Q_{\zeta, i_j-1}, Q_{\zeta, i_j}) \).

The signing algorithm and the verifying algorithm are as follows:  

\[ \text{FMSIG.sign}(i, SK_i, M) \]

Begin

Parse \( < i := i_{i_2},...,i_{d} >, SK_i = \{S_{i_j}, \text{Set}_{i_j} \} \).

Select \( r^{(j)} \in \mathbb{Z}_q^* \), compute and broadcast \( U^{(j)} = r^{(j)} P \).

And then compute \( U = \sum_{j=1}^{n} U^{(j)} \), \( V^{(j)} = s^{(j)}_\zeta H_i(i_{i_2},...,i_{d}) \parallel M, U^{(j)} \) and \( V = \sum_{j=1}^{n} V^{(j)} \).

The final signature is \( < i, \text{sign} = (U, V, \zeta, M_{i_j}) > \)

End

\[ \text{FMSIG.verify}(M, PK, < i, \text{sign} >) \]

Begin

Parse \( < i := i_{i_2},...,i_{d} >, \text{sign} = (U, V, M_{i_j}) \), and \( M_{\zeta, i_j} = (Q_\zeta, \ldots, Q_{\zeta, i_j-1}, Q_{\zeta, i_j}) \),

Verify:

\[ \hat{e}(P, V) = \hat{e}(Q + \sum_{j=1}^{n} H_i(i_{i_2},...,i_{d}) Q_{\zeta, i_j} H_2(Q)) \cdot \hat{e}(U, H_i(i_{i_2},...,i_{d}) \parallel M, U)) \]

If it holds, return “valid”, otherwise, return “invalid.”

End

If the signature \( < i, \text{sign} > \) is correct, above equation can pass the verification, since:

\[ \hat{e}(P, V) = \hat{e}(P, \sum_{j=1}^{n} V^{(j)}) \]

\[ = \hat{e}(P, \sum_{j=1}^{n} (S_{i_j}^{(j)} + r^{(j)} H_i(i_{i_2},...,i_{d}) \parallel M, U^{(j)})) \]

\[ = \hat{e}(P, \sum_{j=1}^{n} (S_{i_j}^{(j)} + r^{(j)} H_i(i_{i_2},...,i_{d}) \parallel M, U^{(j)})) \]

\[ = \hat{e}(H_2(Q), \sum_{j=1}^{n} (Q^{(j)} + \sum_{d=1}^{r} q^{(j)} H_i(i_{i_2},...,i_{d}) Q^{(j)}(i_{j}))) \cdot \hat{e}(P, \sum_{j=1}^{n} r^{(j)} H_i(i_{i_2},...,i_{d}) \parallel M, U)) \]

\[ \hat{e}(H_2(Q), \sum_{j=1}^{n} (Q^{(j)} + \sum_{d=1}^{r} q^{(j)} H_i(i_{i_2},...,i_{d}) Q^{(j)}(i_{j}))) \cdot \hat{e}(P, \sum_{j=1}^{n} r^{(j)} H_i(i_{i_2},...,i_{d}) \parallel M, U)) \]

D. The proposed forward-secure threshold signature scheme

In order to construct the forward-secure threshold scheme, we modify the key algorithm and update algorithm as follows:

(1) The first modification is in key algorithm. All signer \( j=1,\ldots,n \) use Joint-Exp-RSS protocol in [36] to generate \( Q = s_\zeta P \), and \( Q \) is included into the public key.

(2) The second modification is in key algorithm and update algorithm. We consider all the operations as the following stations:

Select \( s_\zeta \in \mathbb{Z}_q^* \), and compute \( Q_\zeta = s_\zeta P \), and \( S_{\zeta} = S_{\zeta, i_j} + s_\zeta H_i(\zeta, Q_\zeta) H_2(Q) \) for some \( \zeta(\zeta = d) \).

We modify the procedure as follows:

All signer \( j=1,\ldots,n \) use Joint-Exp-RSS protocol in [36] to generate \( Q_\zeta = s_\zeta P \). The public commits include \( Q_\zeta \) and \( Q_{\zeta, j}^{(j)} = j=1,\ldots,n \) and signer \( j=1,\ldots,n \) hold \( S_{\zeta}^{(j)} \).
Any signer compute \( Q_s = \sum_{s=1}^{n} C_{b_s} Q_s^{(j)} \), where \( C_{b_s} \) are the computable Lagrange interpolation coefficient. Compute \( S_s^{(j)} = S_{s_s^{(j)}} + s_{s_s^{(j)}} H_s(c_s, Q_s) H_s(Q_s) \).

Set \( M_{s_s^{(j)}} = M_{s_s} = (Q_s, \ldots, Q_{s_{s_{s_s}}}, Q_{s_{s_{s_s}}}) \).

The signing algorithm and the verifying algorithm are as follows:

\[ \text{FBSIG.sign}(i, SK_i, M) \]

\[ \begin{align*}
& \text{Begin} \\
& \quad \text{Parse } < i > = i_1 \ldots i_j, \quad SK_i = \{ S_{s_s^{(j)}}, S_{s_{s_{s_s}}}, M_{s_s^{(j)}} \} \\
& \quad \text{Select } r^{(j)} \in_{R} Z_q^*, \text{ use Joint-Exp-RSS protocol to generate } U = rP. \quad \text{The public commits include } U \text{ and } U^{(j)}, \quad j = 1, \ldots, n. \quad \text{Signer } i(j = 1, \ldots, n) \text{ hold } r^{(j)}. \\
& \quad \text{And then compute } V^{(j)} = S_{s_s^{(j)}} + r^{(j)} H_s(i_1 \ldots i_j || M, U). \\
& \quad \text{Any signer } j \text{ who pass above verification compute } V = \sum_{j=1}^{n} C_{b_s} V^{(j)}. \\
& \quad \text{The final signature is } < i, sign = (U, V, M_{s_s^{(j)}}) >. \\
& \text{End}
\]

\[ \text{FBSIG.verify}(M, PK, < i, sign >) \]

\[ \begin{align*}
& \text{Begin} \\
& \quad \text{Parse } < i > = i_1 \ldots i_j, \quad sign = (U, V, M_{s_s^{(j)}}), \\
& \quad \text{and } M_{s_s^{(j)}} = (Q_s, \ldots, Q_{s_{s_{s_s}}}, Q_{s_{s_{s_s}}}) \\
& \quad \text{Verify:} \\
& \quad \quad \hat{\epsilon}(P, V) = \hat{\epsilon}(P + \sum_{j=1}^{l} H_s(i_1 \ldots i_j, Q_{s_{s_{s_s}}} Q_{s_{s_{s_s}}}, H_s(Q)) \\
& \quad \quad \quad \quad \hat{\epsilon}(U, H_s(i_1 \ldots i_j || M, U)) \\
& \quad \quad \text{If it holds, return “valid”, otherwise, return “invalid”.} \\
& \text{End}
\]

If the signature \(< i, sign >\) is correct, above equation can pass the verification, since:

\[ \hat{\epsilon}(P, V) = \hat{\epsilon}(P + \sum_{j=1}^{l} C_{b_s} V^{(j)}) = \hat{\epsilon}(P, S_{s_s^{(j)}} + r^{(j)} H_s(i_1 \ldots i_j || M, U)) = \hat{\epsilon}(P, S_{s_s^{(j)}} + r^{(j)} H_s(i_1 \ldots i_j, Q_{s_{s_{s_s}}} Q_{s_{s_{s_s}}}, H_s(Q)) + \sum_{j=1}^{l} C_{b_s} r^{(j)} H_s(i_1 \ldots i_j || M, U)) = \hat{\epsilon}(H_s(Q), \sum_{j=1}^{l} C_{b_s} (s_{s_s^{(j)}} H_s(Q) + \sum_{j=1}^{l} C_{b_s} H_s(i_1 \ldots i_j, Q_{s_{s_{s_s}}} Q_{s_{s_{s_s}}}, P)) + \sum_{j=1}^{l} C_{b_s} r^{(j)} H_s(i_1 \ldots i_j || M, U)) = \hat{\epsilon}(H_s(Q), s_{s_s} P + \sum_{j=1}^{l} C_{b_s} H_s(i_1 \ldots i_j, Q_{s_{s_{s_s}}} Q_{s_{s_{s_s}}}, P) + \sum_{j=1}^{l} C_{b_s} r^{(j)} H_s(i_1 \ldots i_j || M, U)) = \hat{\epsilon}(H_s(Q), s_{s_s} P + \sum_{j=1}^{l} C_{b_s} H_s(i_1 \ldots i_j, Q_{s_{s_{s_s}}} Q_{s_{s_{s_s}}}, P) + \sum_{j=1}^{l} C_{b_s} r^{(j)} H_s(i_1 \ldots i_j || M, U)) = \hat{\epsilon}(U, H_s(i_1 \ldots i_j || M, U)) \]

E. The Proposed Forward-Secure Blind Signature Scheme

We use the same key algorithm and update algorithm to generate the public key and the secret key in time \( i \). The signing algorithm and the verifying algorithm are as follows:

\[ \text{FBSIG.sign}(i, SK_i, M) \]

\[ \begin{align*}
& \text{Begin} \\
& \quad \text{Parse } < i > = i_1 \ldots i_j, \quad SK_i = \{ S_{s_s^{(j)}}, S_{s_{s_{s_s}}}, M_{s_s^{(j)}} \} \\
& \quad \text{Firstly, the signer first selects } r \in_{R} Z_q^*, \text{ and computes } U = rP. \\
& \quad \text{And then the signer sends } U \text{ to the requester.} \\
& \quad \text{Blind: Select } x \in_{R} Z_q^*, \text{ and compute } R = H_s(i_1 \ldots i_j || M, U) + xP. \\
& \quad \text{Send } R \text{ to the signer} \\
& \quad \text{Sign: Compute } V' = S_{s_s^{(j)}}, rP. \\
& \quad \text{Return } < i, sign = (U, V', M_{s_s^{(j)}}) > \text{ to the requester.} \\
& \text{End}
\]

\[ \text{FBSIG.verify}(M, PK, < i, sign >) \]

\[ \begin{align*}
& \text{Begin} \\
& \quad \text{Parse } < i > = i_1 \ldots i_j, \quad sign = (U, V', M_{s_s^{(j)}}), \\
& \quad \text{and } M_{s_s^{(j)}} = (Q_s, \ldots, Q_{s_{s_{s_s}}}, Q_{s_{s_{s_s}}}) \\
& \quad \text{Verify:} \\
& \quad \quad \hat{\epsilon}(P, V') = \hat{\epsilon}(P + \sum_{j=1}^{l} H_s(i_1 \ldots i_j, Q_{s_{s_{s_s}}} Q_{s_{s_{s_s}}}, H_s(Q)) + \sum_{j=1}^{l} C_{b_s} r^{(j)} H_s(i_1 \ldots i_j || M, U)) = \hat{\epsilon}(U, H_s(i_1 \ldots i_j || M, U)) = \hat{\epsilon}(P, R H_s(i_1 \ldots i_j || M, U)) = \hat{\epsilon}(P, S_{s_s^{(j)}}, rP, H_s(i_1 \ldots i_j || M, U)) = \hat{\epsilon}(P, s_s H_s(Q) + \sum_{j=1}^{l} s_{s_{s_s}} H_s(i_1 \ldots i_j, Q_{s_{s_{s_s}}}, Q_{s_{s_{s_s}}}, H_s(Q)) + \sum_{j=1}^{l} C_{b_s} r^{(j)} H_s(i_1 \ldots i_j || M, U)) = \hat{\epsilon}(P, rP, H_s(i_1 \ldots i_j || M, U)) = \hat{\epsilon}(P, s_s H_s(Q) + \sum_{j=1}^{l} s_{s_{s_s}} H_s(i_1 \ldots i_j, Q_{s_{s_{s_s}}}, Q_{s_{s_{s_s}}}, P, H_s(Q)) + \sum_{j=1}^{l} C_{b_s} r^{(j)} H_s(i_1 \ldots i_j || M, U)) = \hat{\epsilon}(P, rP, H_s(i_1 \ldots i_j || M, U)) = \hat{\epsilon}(P, s_s P + \sum_{j=1}^{l} s_{s_{s_s}} H_s(i_1 \ldots i_j, Q_{s_{s_{s_s}}}, Q_{s_{s_{s_s}}}, P, H_s(Q)) + \sum_{j=1}^{l} C_{b_s} r^{(j)} H_s(i_1 \ldots i_j || M, U)) = \hat{\epsilon}(P, U, H_s(i_1 \ldots i_j || M, U)) \\
& \text{If it holds, return “valid”, otherwise, return “invalid”.} \\
& \text{End}
\]
Adding forward security to signatures is an effective method to deal with the key exposure problem. In this paper, we construct two forward-secure multisignature schemes one forward-secure threshold signature scheme, and one forward-secure blind signature scheme. Such schemes can be applied to many e-commerce applications.

ACKNOWLEDGMENT

This research is supported by National Natural Science Foundation of China (60703089), the Science and Technology Project of Provincial Education Department of Shandong (J08LJ02), the Scientific Research Foundation for The Excellent Middle-aged and Young Scientists of Shandong Province of China (2008BS01011), National Cryptologic Development Foundation of China, and the Youth Research Foundation of Qingdao University.

REFERENCES


Fanyu Kong was born in China in 1976. He received the BS, MS, and PhD degrees in computer science from Shandong University, Shandong, China, in 2000, 2003, and 2006, respectively.

He became a lecturer, an associate professor of computer science in the College of Information Engineering at Qingdao University, China, in 2006 and 2007, respectively. He is currently an associate professor in the College of Information Engineering at Qingdao University, China. His research interests include encryption, digital signature, cryptographic protocol and network security.

Jia Yu was born in China in 1976. He received the BS, MS, and PhD degrees in computer science from Shandong University, Shandong, China, in 2000, 2003, and 2006, respectively.

He became a lecturer, an associate professor of computer science in the College of Information Engineering at Qingdao University, China, in 2006 and 2007, respectively. He is currently an associate professor in the College of Information Engineering at Qingdao University, China. His research interests include encryption, digital signature, cryptographic protocol and network security.

Fanyu Kong currently is a member of Chinese Association for cryptologic Research and Chinese Computer Federation.

Jia Yu currently is a member of Chinese Association for cryptologic Research.

Yongkui Chen was born in China in 1984. He received BS degrees from Xian dianzi University, China, in 2006.

His research interests include digital signature, cryptographic protocol.

Dr. Chen currently is a member of Chinese Association for cryptologic Research.

Jia Yu was born in China in 1976. He received the BS, MS, and PhD degrees in computer science from Shandong University, Shandong, China, in 2000, 2003, and 2006, respectively.

He became a lecturer, an associate professor of computer science in the College of Information Engineering at Qingdao University, China, in 2006 and 2007, respectively. He is currently an associate professor in the College of Information Engineering at Qingdao University, China. His research interests include encryption, digital signature, cryptographic protocol and network security.

Dr. Yu currently is a member of Chinese Association for cryptologic Research and Chinese Computer Federation.

Guowen Li was born in China in 1976. He received the BS, MS, and PhD degrees in computer science from Shandong University, Shandong, China, in 1998 and 2006, respectively.

She became a lecturer of computer science in the College of Information Engineering at Qingdao University, China, in 2006. She is currently a fellow in the College of Information Engineering at Qingdao University, China. Her research interests include cryptography and network security.

Dr. Li currently is a member of Chinese Association for cryptologic Research.


Fanyu Kong was born in China in 1976. He received the BS, MS, and PhD degrees in computer science from Shandong University, Shandong, China, in 2000, 2003, and 2006, respectively.

He became a lecturer, an associate professor of computer science in the College of Information Engineering at Qingdao University, China, in 2006 and 2007, respectively. He is currently an associate professor in the College of Information Engineering at Qingdao University, China. His research interests include encryption, digital signature, cryptographic protocol and network security.

Dr. Yu currently is a member of Chinese Association for cryptologic Research and Chinese Computer Federation.

Fanyu Kong was born in China in 1976. He received the BS, MS, and PhD degrees in computer science from Shandong University, Shandong, China, in 2000, 2003, and 2006, respectively.

He became a lecturer, an associate professor of computer science in the College of Information Engineering at Qingdao University, China, in 2006 and 2007, respectively. He is currently an associate professor in the College of Information Engineering at Qingdao University, China. His research interests include encryption, digital signature, cryptographic protocol and network security.

Dr. Yu currently is a member of Chinese Association for cryptologic Research and Chinese Computer Federation.


Fanyu Kong was born in China in 1976. He received the BS, MS, and PhD degrees in computer science from Shandong University, Shandong, China, in 2000, 2003, and 2006, respectively.

He became a lecturer, an associate professor of computer science in the College of Information Engineering at Qingdao University, China, in 2006 and 2007, respectively. He is currently an associate professor in the College of Information Engineering at Qingdao University, China. His research interests include encryption, digital signature, cryptographic protocol and network security.

Dr. Yu currently is a member of Chinese Association for cryptologic Research and Chinese Computer Federation.


Fanyu Kong was born in China in 1976. He received the BS, MS, and PhD degrees in computer science from Shandong University, Shandong, China, in 2000, 2003, and 2006, respectively.

He became a lecturer, an associate professor of computer science in the College of Information Engineering at Qingdao University, China, in 2006 and 2007, respectively. He is currently an associate professor in the College of Information Engineering at Qingdao University, China. His research interests include encryption, digital signature, cryptographic protocol and network security.

Dr. Yu currently is a member of Chinese Association for cryptologic Research and Chinese Computer Federation.


Fanyu Kong was born in China in 1976. He received the BS, MS, and PhD degrees in computer science from Shandong University, Shandong, China, in 2000, 2003, and 2006, respectively.

He became a lecturer, an associate professor of computer science in the College of Information Engineering at Qingdao University, China, in 2006 and 2007, respectively. He is currently an associate professor in the College of Information Engineering at Qingdao University, China. His research interests include encryption, digital signature, cryptographic protocol and network security.

Dr. Yu currently is a member of Chinese Association for cryptologic Research and Chinese Computer Federation.


Fanyu Kong was born in China in 1976. He received the BS, MS, and PhD degrees in computer science from Shandong University, Shandong, China, in 2000, 2003, and 2006, respectively.

He became a lecturer, an associate professor of computer science in the College of Information Engineering at Qingdao University, China, in 2006 and 2007, respectively. He is currently an associate professor in the College of Information Engineering at Qingdao University, China. His research interests include encryption, digital signature, cryptographic protocol and network security.

Dr. Yu currently is a member of Chinese Association for cryptologic Research and Chinese Computer Federation.


Fanyu Kong was born in China in 1976. He received the BS, MS, and PhD degrees in computer science from Shandong University, Shandong, China, in 2000, 2003, and 2006, respectively.

He became a lecturer, an associate professor of computer science in the College of Information Engineering at Qingdao University, China, in 2006 and 2007, respectively. He is currently an associate professor in the College of Information Engineering at Qingdao University, China. His research interests include encryption, digital signature, cryptographic protocol and network security.

Dr. Yu currently is a member of Chinese Association for cryptologic Research and Chinese Computer Federation.