Forward-Secure Multisignature, Threshold Signature and Blind Signature Schemes

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

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

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

Guowen Li
[3]School 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

doi:10.4304/jnw.5.6.634-641

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 published in [32-34]. Forward-secure blind signatures were proposed in [30, 35].

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 in [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 $\hat{e}: G_1 \times G_1 \to G_2$ satisfies:

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

Definition 1 (CDH Assumption). A probabilistic algorithm *A* is said (t, ε)-attack CDH problem in G_1 if *A* runs at most time *t*, computes CDH problem with an advantage of at least ε. We say that G_1 is a (t, ε)-secure CDH group if no probabilistic algorithm *A* (t, ε)-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 *FMSIG*=(***FMSIG.key, FMSIG.update, FMSIG.sign, FMSIG.verify***), where:

1. ***FMSIG.key***: the key generation algorithm, is a probabilistic algorithm which takes as input a security parameter $k \in N$ and the total number of time periods *T*, and generates a public key PK and the initial secret key $SK_0^{(j)}$ for signer j(j=1,…,n).
2. ***FMSIG.update***: the key update algorithm, is a probabilistic algorithm which takes as input the secret key $SK_i^{(j)}$ signer *j* holds of the current period *i* and generates the new secret key $SK_{i+1}^{(j)}$ for the next period.
3. ***FMSIG.sign***: the signing algorithm, takes as input the secret key $SK_i^{(j)}$ 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 $< i, sign >$ 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 $< i, sign >$, and output 1 when $< i, sign >$ 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 *FTSIG*=(***FTSIG.key, FTSIG.update, FTSIG.sign, FTSIG.verify***), where:

1. ***FMSIG.key***: the key generation algorithm, is a probabilistic algorithm which takes as input a security parameter $k \in N$ and the total number of time periods *T*, and generates a public key PK and the initial secret key $SK_0^{(j)}$ for player *j*(*j*=1,…,n).
2. ***FMSIG.update***: the key update algorithm, is a probabilistic algorithm which takes as input all the secret key $SK_i^{(j)}$ players *j*(*j*=1,…,n) hold of the current period *i* and generates the new secret key $SK_{i+1}^{(j)}$ for the next period.
3. ***FTSIG.sign***: the signing algorithm, takes as input all the secret key $SK_i^{(j)}$ players *j*(*j*=1,…,n) hold of the

current time period i and a message M, and any t players generate the final threshold signature $< i, sign >$ 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 $< i, sign >$, and output 1 when $< i, sign >$ 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 *FBSIG*=(***FBSIG.key, FBSIG.update, FBSIG.sign, FBSIG.verify***), where:

1. ***FBSIG.key***: the key generation algorithm, is a probabilistic algorithm which takes as input a security parameter $k \in N$ 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 σ'
   (3)Unblind: On a blind signature σ' and random string r as the input, it outputs the final unblind signature $< i, sign >$.
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 $< i, sign >$, and output 1 when $< i, sign >$ 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 $FSIG.key$ (k,l,T)

Begin
  Run IG(1^k) to generate additive group G_1 and multiplicative group G_2 with same prime order q and an admissible pairing $\hat{e}: G_1 \times G_1 \to G_2$.
  Select cryptographic hash functions $H_1: \{0,1\}^* \times G_1 \to Z_q^*$, $H_2: G_1 \to G_1$, and $H_3: \{0,1\}^* \times G_1 \to G_1$.
  Select generator $P \in_R G_1$ and secret $s_\varepsilon \in_R Z_q^*$, and let $Q = s_\varepsilon P$, $S_\varepsilon = s_\varepsilon H_2(Q)$.
  Let the public key be $PK = \{G_1, G_2, \hat{e}, H_1, H_2, H_3, P, Q\}$.
  Select $s_0, s_1 \in_R Z_q^*$, and compute $Q_0 = s_0 P,\ Q_1 = s_1 P$.
  Compute $S_0 = S_\varepsilon + s_0 H_1(0, Q_0) H_2(Q)$
  and $S_1 = S_\varepsilon + s_1 H_1(1, Q_1) H_2(Q)$.
  For $j = 1$ to $l-1$
  {
    Select $s_{0^j 0}, s_{0^j 1} \in_R Z_q^*$,
    and compute $Q_{0^j 0} = s_{0^j 0} P,\ Q_{0^j 1} = s_{0^j 1} P$.
    Compute $S_{0^j 0} = S_{0^j} + s_{0^j 0} H_1(0^j 0, Q_{0^j 0}) H_2(Q)$,
    and $S_{0^j 1} = S_{0^j} + s_{0^j 1} H_1(0^j 1, Q_{0^j 1}) H_2(Q)$.
  }
  Set $SK_0 = \{S_{0^l}, Set_{0^l}, M_{0^l}\}$,
  where $M_{0^l} = (Q_0, ..., Q_{0^{l-1}}, Q_{0^l})$, and
  $Set_{0^l} = (< S_1, Q_1 >, < S_{01}, Q_{01} >, ..., < S_{0^{l-1}1}, Q_{0^{l-1}1} >)$.
  Erase all interim data, and return PK, SK_0.
End.

(2) algorithm $FSIG.update(SK_i)$

Begin
  If $i = T-1$ then
    Let $SK_T = \phi$.
  Else
  {
    Parse $< i >= i_0 i_1 i_2 ... i_l$, $(i_0 = \varepsilon)$,
    $SK_i = \{S_{<i>}, Set_{<i>}, M_{<i>}\}$,
    and $M_{<i>} = (Q_{i_1}, ..., Q_{i_1 i_2 ... i_{l-1}}, Q_{<i>})$.
    If $i_l = 0$ then
    {
      Find $< S_{<i+1>}, Q_{<i+1>} >$ from $Set_{<i>}$,

and set $Set_{<i+1>} = Set_{<i>} - \{< S_{<i+1>}, Q_{<i+1>} >\}$.
Set $M_{<i+1>} = (M_{<i>} - \{Q_{<i>}\}) \cup \{Q_{<i+1>}\}$.
Set $SK_{i+1} = (S_{<i+1>}, Set_{<i+1>}, M_{<i+1>})$.
}
Else
{
Find the maximal $j(1 \le j < l)$ satisfying $i_j = 0$,
and let $\eta = i_0 i_1 ... i_{j-1} 1,\ (i_0 = \varepsilon)$.
Here $< i+1 >= \eta 0^{l-j}$.
Find $< S_\eta, Q_\eta >$ from $Set_{<i>}$,
Set $M_{<i+1>} = M_{<i>} \cup \{Q_\eta\}$,
and set $Set_{<i+1>} = Set_{<i>} - \{< S_\eta, Q_\eta >\}$.
Set $M_{<i+1>} = M_{<i+1>} - \{Q_{i_1 i_2 ... i_j}, Q_{i_1 i_2 ... i_j 1}, ...,$
$Q_{i_1 i_2 ... i_j 1^{l-j-1}}, Q_{<i>}\}$.
For $m = 1$ to $l - j$
{
Select $s_{\eta 0^m}, s_{\eta 0^{m-1} 1} \in_R Z_q$,
and compute $Q_{\eta 0^m} = s_{\eta 0^m} P$,
$Q_{\eta 0^{m-1} 1} = s_{\eta 0^{m-1} 1} P$,
$S_{\eta 0^m} = S_{\eta 0^{m-1}} + s_{\eta 0^m} H_1(\eta 0^m, Q_{\eta 0^m}) H_2(Q)$,
and $S_{\eta 0^{m-1} 1} = S_{\eta 0^{m-1}} + s_{\eta 0^{m-1} 1} H_1(\eta 0^{m-1} 1$
$, Q_{\eta 0^{m-1} 1}) H_2(Q)$.
Set $Set_{<i+1>} = Set_{<i+1>} \cup \{< S_{\eta 0^{m-1} 1}, Q_{\eta 0^{m-1} 1} >\}$,
and $M_{<i+1>} = M_{<i+1>} \cup \{Q_{\eta 0^m}\}$.
}
Set $SK_{i+1} = (S_{<i+1>}, Set_{<i+1>}, M_{<i+1>})$.
}
}
Erase all interim data, and return SK_{i+1}.
End

(3) algorithm $FSIG.sign(i, SK_i, M)$
Begin
Parse $< i >= i_1 i_2 ... i_l$, $SK_i = \{S_{<i>}, Set_{<i>}, M_{<i>}\}$.
Select $r \in_R Z_q^*$, and compute $U = rP$.
Compute $V = S_{<i>} + rH_3(i_1 i_2 ... i_l \| M, U)$.
Return signature $< i, sign = (U, V, M_{<i>}) >$.
End

(4) algorithm $FSIG.verify(M, PK, < i, sign >)$
Begin
Parse $< i >= i_1 i_2 ... i_l$, $sign = (U, V, M_{<i>})$,
and $M_{<i>} = (Q_{i_1}, ..., Q_{i_1 i_2 ... i_{l-1}}, Q_{<i>})$.
Verify:

$$\hat{e}(P, V) \overset{?}{=} \hat{e}(Q + \sum_{j=1}^{l} H_1(i_1 i_2 ... i_j, Q_{i_1 i_2 ... i_j}) Q_{i_1 i_2 ... i_j}, H_2(Q)) \cdot \hat{e}(U, H_3(i_1 i_2 ... i_l \| M, U))$$

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* $SK_i^{(j)} = \{S_{<i>}^{(j)}, Set_{<i>}^{(j)}, M_{<i>}^{(j)}\}$, where $M_{<i>}^{(j)} = (Q_{i_1}^{(j)}, ..., Q_{i_1 i_2 ... i_{l-1}}^{(j)}, Q_{<i>}^{(j)})$.

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

$FMSIG.sign(i, SK_i, M)$
Begin
Parse $< i >= i_1 i_2 ... i_l$, $SK_i^{(j)} = \{S_{<i>}^{(j)}, Set_{<i>}^{(j)}, M_{<i>}^{(j)}\}$.
Select $r^{(j)} \in_R Z_q^*$, and compute $U^{(j)} = r^{(j)} P$.
Compute $V^{(j)} = S_{<i>}^{(j)} + r^{(j)} H_3(i_1 i_2 ... i_l \| M, U^{(j)})$.
The partial signature is $< i, sign = (U^{(j)}, V^{(j)}, M_{<i>}^{(j)}) >$.
We can use the following equation to verify whether the partial signature is valid or not:

$$\hat{e}(P, V^{(j)}) \overset{?}{=} \hat{e}(Q^{(j)} + \sum_{d=1}^{l} H_1(i_1 i_2 ... i_d, Q_{i_1 i_2 ... i_d}^{(j)}) Q_{i_1 i_2 ... i_d}^{(j)}, H_2(Q^{(j)})) \cdot \hat{e}(U^{(j)}, H_3(i_1 i_2 ... i_l \| M, U^{(j)}))$$

Compute $V = \sum_{j=1}^{n} V^{(j)}$.
Let $\xi_{<i>} = \{M_{<i>}^{(1)}, ..., M_{<i>}^{(n)}\}$
The final signature is $< i, sign = (\{U^{(j)}\}_{j=1,...,n}, V, \xi_{<i>}) >$
End

$FMSIG.verify(M, PK, < i, sign >)$
Begin
Parse $< i >= i_1 i_2 ... i_l$, $sign = (\{U^{(j)}\}_{j=1,...,n}, V, \xi_{<i>})$,
and $\xi_{<i>} = \{M_{<i>}^{(1)}, ..., M_{<i>}^{(n)}\}$,
where $M_{<i>}^{(j)} = (Q_{i_1}^{(j)}, ..., Q_{i_1 i_2 ... i_{l-1}}^{(j)}, Q_{<i>}^{(j)})$.
Verify:

$$\hat{e}(P, V) \overset{?}{=} \prod_{j=1}^{n} \hat{e}(Q^{(j)} + \sum_{d=1}^{l} H_1(i_1 i_2 ... i_d, Q_{i_1 i_2 ... i_d}^{(j)}) Q_{i_1 i_2 ... i_d}^{(j)}, H_2(Q^{(j)})) \cdot \prod_{j=1}^{n} \hat{e}(U^{(j)}, H_3(i_1 i_2 ... i_l \| M, U^{(j)}))$$

If it holds, return "valid", otherwise, return "invalid".
End

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

$$
\begin{aligned}
&\hat{e}(P,V) \\
&= \hat{e}(P, \textstyle\sum_{j=1}^{n} V^{(j)}) \\
&= \textstyle\prod_{j=1}^{n} \hat{e}(P, V^{(j)}) \\
&= \textstyle\prod_{j=1}^{n} \hat{e}(P, S_{<i>}^{(j)} + r^{(j)} H_3(i_1 i_2 \dots i_l \parallel M, U^{(j)})) \\
&= \textstyle\prod_{j=1}^{n} \hat{e}(P, S_{<i>}^{(j)}) \cdot \prod_{j=1}^{n} \hat{e}(P, r^{(j)} H_3(i_1 i_2 \dots i_l \parallel M, U^{(j)})) \\
&= \textstyle\prod_{j=1}^{n} \hat{e}(P, s_{\varepsilon}^{(j)} H_2(Q^{(j)}) + \sum_{d=1}^{l} s_{i_1 i_2 \dots i_d}^{(j)} H_1(i_1 i_2 \dots i_d, Q_{i_1 i_2 \dots i_d}^{(j)}) \\
&\quad H_2(Q^{(j)})) \cdot \textstyle\prod_{j=1}^{n} \hat{e}(r^{(j)} P, H_3(i_1 i_2 \dots i_l \parallel M, U^{(j)})) \\
&= \textstyle\prod_{j=1}^{n} \hat{e}(H_2(Q^{(j)}), s_{\varepsilon}^{(j)} P + \sum_{d=1}^{l} s_{i_1 i_2 \dots i_d}^{(j)} H_1(i_1 i_2 \dots i_d, Q_{i_1 i_2 \dots i_d}^{(j)}) P) \\
&\quad \cdot \textstyle\prod_{j=1}^{n} \hat{e}(U^{(j)}, H_3(i_1 i_2 \dots i_l \parallel M, U^{(j)})) \\
&= \textstyle\prod_{j=1}^{n} \hat{e}(Q^{(j)} + \sum_{d=1}^{l} H_1(i_1 i_2 \dots i_d, Q_{i_1 i_2 \dots i_d}^{(j)}) Q_{i_1 i_2 \dots i_d}^{(j)}, H_2(Q^{(j)})) \\
&\quad \cdot \textstyle\prod_{j=1}^{n} \hat{e}(U^{(j)}, H_3(i_1 i_2 \dots i_l \parallel M, U^{(j)}))
\end{aligned}
$$

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(j=1,…,n) compute $Q^{(j)} = s_{\varepsilon}^{(j)} P$ and $Q = \sum_{j=1}^{n} Q^{(j)}$. The public key is $PK = \{G_1, G_2, \hat{e}, H_1, H_2, H_3, P, Q\}$.

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

Select $s_{\zeta} \in_R Z_q^*$, and compute $Q_{\zeta} = s_{\zeta} P$, and $S_{\zeta} = S_{\zeta|_{d-1}} + s_{\zeta} H_1(\zeta, Q_{\zeta}) H_2(Q)$ for some $\zeta(|\zeta| = d)$.

We modify the procedure as follows:

All signer j(j=1,…,n) select $s_{\zeta}^{(j)} \in_R Z_q^*$, and compute and broadcast $Q_{\zeta}^{(j)} = s_{\zeta}^{(j)} P$, and then compute $Q_{\zeta} = \sum_{j=1}^{n} Q_{\zeta}^{(j)}$ and $S_{\zeta}^{(j)} = S_{\zeta|_{d-1}}^{(j)} + s_{\zeta}^{(j)} H_1(\zeta, Q_{\zeta}) H_2(Q)$.

Set $M_{<i>}^{(j)} = M_{<i>} = (Q_{i_1}, \dots, Q_{i_1 i_2 \dots i_{l-1}}, Q_{<i>})$.

The signing algorithm and the verifying algorithm are as follows:

$FMSIG.sign(i, SK_i, M)$

Begin

Parse $<i> = i_1 i_2 \dots i_l$, $SK_i^{(j)} = \{S_{<i>}^{(j)}, Set_{<i>}^{(j)}, \square_{<i>}^{(j)}\}$.

Select $r^{(j)} \in_R Z_q^*$, compute and broadcast $U^{(j)} = r^{(j)} P$.

And then compute $U = \sum_{j=1}^{n} U^{(j)}$, $V^{(j)} = S_{<i>}^{(j)} + r^{(j)} H_3(i_1 i_2 \dots i_l \parallel M, U)$ and $V = \sum_{j=1}^{n} V^{(j)}$.

The final signature is $<i, sign = (U, V, \square_{<i>})>$

End

$FMSIG.verify(M, PK, <i, sign>)$

Begin

Parse $<i> = i_1 i_2 \dots i_l$, $sign = (U, V, M_{<i>})$,

and $M_{<i>} = (Q_{i_1}, \dots, Q_{i_1 i_2 \dots i_{l-1}}, Q_{<i>})$,

Verify:

$$
\begin{aligned}
\hat{e}(P,V) &\overset{?}{=} \hat{e}(Q + \textstyle\sum_{j=1}^{l} H_1(i_1 i_2 \dots i_j, Q_{i_1 i_2 \dots i_j}) Q_{i_1 i_2 \dots i_j}, H_2(Q)) \\
&\quad \cdot \hat{e}(U, H_3(i_1 i_2 \dots i_l \parallel M, U))
\end{aligned}
$$

If it holds, return "valid", otherwise, return "invalid".

End

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

$$
\begin{aligned}
&\hat{e}(P,V) \\
&= \hat{e}(P, \textstyle\sum_{j=1}^{n} V^{(j)}) \\
&= \hat{e}(P, \textstyle\sum_{j=1}^{n} (S_{<i>}^{(j)} + r^{(j)} H_3(i_1 i_2 \dots i_l \parallel M, U))) \\
&= \hat{e}(P, \textstyle\sum_{j=1}^{n} (s_{\varepsilon}^{(j)} H_2(Q) + \sum_{d=1}^{l} s_{i_1 i_2 \dots i_d}^{(j)} H_1(i_1 i_2 \dots i_d, Q_{i_1 i_2 \dots i_d}) H_2(Q)) \\
&\quad + \textstyle\sum_{j=1}^{n} r^{(j)} H_3(i_1 i_2 \dots i_l \parallel M, U)) \\
&= \hat{e}(H_2(Q), \textstyle\sum_{j=1}^{n} (s_{\varepsilon}^{(j)} P + \sum_{d=1}^{l} s_{i_1 i_2 \dots i_d}^{(j)} H_1(i_1 i_2 \dots i_d, Q_{i_1 i_2 \dots i_d}) P)) \\
&\quad \cdot \hat{e}(P, \textstyle\sum_{j=1}^{n} r^{(j)} H_3(i_1 i_2 \dots i_l \parallel M, U)) \\
&= \hat{e}(H_2(Q), \textstyle\sum_{j=1}^{n} (Q^{(j)} + \sum_{d=1}^{l} H_1(i_1 i_2 \dots i_d, Q_{i_1 i_2 \dots i_d}) Q_{i_1 i_2 \dots i_d}^{(j)})) \\
&\quad \cdot \hat{e}(\textstyle\sum_{j=1}^{n} r^{(j)} P, H_3(i_1 i_2 \dots i_l \parallel M, U)) \\
&= \hat{e}(H_2(Q), \textstyle\sum_{j=1}^{n} Q^{(j)} + \sum_{j=1}^{n} \sum_{d=1}^{l} H_1(i_1 i_2 \dots i_d, Q_{i_1 i_2 \dots i_d}) Q_{i_1 i_2 \dots i_d}^{(j)}) \\
&\quad \cdot \hat{e}(\textstyle\sum_{j=1}^{n} U^{(j)}, H_3(i_1 i_2 \dots i_l \parallel M, U)) \\
&= \hat{e}(H_2(Q), Q + \textstyle\sum_{d=1}^{l} H_1(i_1 i_2 \dots i_d, Q_{i_1 i_2 \dots i_d}) Q_{i_1 i_2 \dots i_d}) \\
&\quad \cdot \hat{e}(U, H_3(i_1 i_2 \dots i_l \parallel M, U))
\end{aligned}
$$

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(j=1,…,n) use *Joint-Exp-RSS* protocol in [36] to generate $Q = s_{\varepsilon} P$, 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 $s_{\zeta} \in_R Z_q^*$, and compute $Q_{\zeta} = s_{\zeta} P$, and $S_{\zeta} = S_{\zeta|_{d-1}} + s_{\zeta} H_1(\zeta, Q_{\zeta}) H_2(Q)$ for some $\zeta(|\zeta| = d)$.

We modify the procedure as follows:

All signer j(j=1,…,n) use *Joint-Exp-RSS* protocol in [36] to generate $Q_{\zeta} = s_{\zeta} P$. The public commits include Q_{ζ} and $Q_{\zeta}^{(j)}$ j(j=1,…,n) and signer j(j=1,…,n) hold $s_{\zeta}^{(j)}$.

Any t signer j compute $Q_\zeta = \sum_{j=1}^{n} C_{Bj} Q_\zeta^{(j)}$, where C_{Bj} are the computable Lagrange interpolation coefficient. Compute $S_\zeta^{(j)} = S_{\zeta|_{d-1}}^{(j)} + s_\zeta^{(j)} H_1(\zeta, Q_\zeta) H_2(Q)$.

Set $M_{<i>}^{(j)} = M_{<i>} = (Q_{i_1}, ..., Q_{i_1 i_2 ... i_{l-1}}, Q_{<i>})$.

The signing algorithm and the verifying algorithm are as follows:

FTSIG.sign(i, SK_i, M)

Begin

Parse $< i >= i_1 i_2 ... i_l$, $SK_i^{(j)} = \{S_{<i>}^{(j)}, Set_{<i>}^{(j)}, M_{<i>}^{(j)}\}$.

Select $r^{(j)} \in_R Z_q^*$, use *Joint-Exp-RSS* protocol to generate $U = rP$. The public commits include U and $U^{(j)}$, j(j=1,…,n). Signer j(j=1,…,n) hold $r^{(j)}$.

And then compute $V^{(j)} = S_{<i>}^{(j)} + r^{(j)} H_3(i_1 i_2 ... i_l \| M, U)$.

Any t signer j who pass above verification compute $V = \sum_{j \in B} C_{Bj} V^{(j)}$.

The final signature is $< i, sign = (U, V, M_{<i>}) >$

End

FTSIG.verify$(M, PK, < i, sign >)$

Begin

Parse $< i >= i_1 i_2 ... i_l$, $sign = (U, V, M_{<i>})$,

and $M_{<i>} = (Q_{i_1}, ..., Q_{i_1 i_2 ... i_{l-1}}, Q_{<i>})$,

Verify:

$$\hat{e}(P, V) \overset{?}{=} \hat{e}(Q + \sum_{j=1}^{l} H_1(i_1 i_2 ... i_j, Q_{i_1 i_2 ... i_j}) Q_{i_1 i_2 ... i_j}, H_2(Q)) \cdot \hat{e}(U, H_3(i_1 i_2 ... i_l \| M, U))$$

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

End

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

$$\begin{aligned}
&\hat{e}(P, V) \\
&= \hat{e}(P, \sum_{j \in B} C_{Bj} V^{(j)}) \\
&= \hat{e}(P, \sum_{j \in B} C_{Bj} (S_{<i>}^{(j)} + r^{(j)} H_3(i_1 i_2 ... i_l \| M, U))) \\
&= \hat{e}(P, \sum_{j \in B} C_{Bj} (s_\varepsilon^{(j)} H_2(Q) + \sum_{d=1}^{l} s_{i_1 i_2 ... i_d}^{(j)} H_1(i_1 i_2 ... i_d, Q_{i_1 i_2 ... i_d}) H_2(Q)) \\
&\quad + \sum_{j \in B} C_{Bj} r^{(j)} H_3(i_1 i_2 ... i_l \| M, U)) \\
&= \hat{e}(H_2(Q), \sum_{j \in B}^{n} C_{Bj} (s_\varepsilon^{(j)} P + \sum_{d=1}^{l} s_{i_1 i_2 ... i_d}^{(j)} H_1(i_1 i_2 ... i_d, Q_{i_1 i_2 ... i_d}) P)) \\
&\quad \cdot \hat{e}(P, \sum_{j \in B} C_{Bj} r^{(j)} H_3(i_1 i_2 ... i_l \| M, U)) \\
&= \hat{e}(H_2(Q), s_\varepsilon P + \sum_{d=1}^{l} s_{i_1 i_2 ... i_d} H_1(i_1 i_2 ... i_d, Q_{i_1 i_2 ... i_d}) P) \\
&\quad \cdot \hat{e}(P, r H_3(i_1 i_2 ... i_l \| M, U)) \\
&= \hat{e}(H_2(Q), Q + \sum_{d=1}^{l} H_1(i_1 i_2 ... i_d, Q_{i_1 i_2 ... i_d}) Q_{i_1 i_2 ... i_d}) \\
&\quad \cdot \hat{e}(U, H_3(i_1 i_2 ... i_l \| M, U))
\end{aligned}$$

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:

FBSIG.sign(i, SK_i, M)

Begin

Parse $< i >= i_1 i_2 ... i_l$, $SK_i = \{S_{<i>}, Set_{<i>}, M_{<i>}\}$.

Firstly, the signer first selects $r \in_R Z_q^*$, and computes $U = rP$.

And then the signer sends U to the requester.

Blind: Select $x \in_R Z_q^*$, and compute

$R = H_3(i_1 i_2 ... i_l \| M, U) + xP$.

Send R to the signer

Sign: Compute $V' = S_{<i>} + rR$.

Return $< i, sign = (U, V', M_{<i>}) >$ to the requester.

Unblind: After receives V' , the requester unblinds it by computing $V = V' - xU$. The final blind signature is $< i, sign = (U, V, M_{<i>}) >$

End

FBSIG.verify$(M, PK, < i, sign >)$

Begin

Parse $< i >= i_1 i_2 ... i_l$, $sign = (U, V, M_{<i>})$,

and $M_{<i>} = (Q_{i_1}, ..., Q_{i_1 i_2 ... i_{l-1}}, Q_{<i>})$.

Verify:

$$\hat{e}(P, V) \overset{?}{=} \hat{e}(Q + \sum_{j=1}^{l} H_1(i_1 i_2 ... i_j, Q_{i_1 i_2 ... i_j}) Q_{i_1 i_2 ... i_j}, H_2(Q)) \cdot \hat{e}(U, H_3(i_1 i_2 ... i_l \| M, U))$$

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

End

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

$$\begin{aligned}
&\hat{e}(P, V) \\
&= \hat{e}(P, V' - xU) \\
&= \hat{e}(P, S_{<i>} + rR - rxP) \\
&= \hat{e}(P, S_{<i>} + rH_3(i_1 i_2 ... i_l \| M, U) + rxP - rxP) \\
&= \hat{e}(P, S_{<i>} + rH_3(i_1 i_2 ... i_l \| M, U)) \\
&= \hat{e}(P, S_{<i>}) \cdot \hat{e}(P, rH_3(i_1 i_2 ... i_l, M, U)) \\
&= \hat{e}(P, s_\varepsilon H_2(Q) + \sum_{j=1}^{l} s_{i_1 i_2 ... i_j} H_1(i_1 i_2 ... i_j, Q_{i_1 i_2 ... i_j}) H_2(Q)) \\
&\quad \cdot \hat{e}(rP, H_3(i_1 i_2 ... i_l \| M, U)) \\
&= \hat{e}(P, (s_\varepsilon + \sum_{j=1}^{l} s_{i_1 i_2 ... i_j} H_1(i_1 i_2 ... i_j, Q_{i_1 i_2 ... i_j})) H_2(Q)) \\
&\quad \cdot \hat{e}(rP, H_3(i_1 i_2 ... i_l \| M, U)) \\
&= \hat{e}(s_\varepsilon P + \sum_{j=1}^{l} s_{i_1 i_2 ... i_j} H_1(i_1 i_2 ... i_j, Q_{i_1 i_2 ... i_j}) P, H_2(Q)) \\
&\quad \cdot \hat{e}(U, H_3(i_1 i_2 ... i_l \| M, U))
\end{aligned}$$

$$= \hat{e}(Q + \sum\nolimits_{j=1}^{l} H_1(i_1 i_2 ... i_j, Q_{i_1 i_2 ... i_j}) Q_{i_1 i_2 ... i_j}, H_2(Q))$$
$$\cdot \hat{e}(U, H_3(i_1 i_2 ... i_l \,\|\, M, U))$$

IV. RELATED ANALYSIS AND DISCUSS

(1) Security

The security of our schemes depends on the computation Diffie-Hellman assumption. The security proof of our schemes can be easy to be modified from the proof in [11]. Here we skip the proof procedure.

(2) Efficiency.

Because our schemes are based on binary tree, they enjoy an advantage, that is, there is no cost parameters including key generation time, key update time, signing time, verifying time, and signature size, public key size, secret storage size has a complexity more than $O(\log T)$ in terms of the total number of time periods T in this scheme.

V. CONCLUSIONS

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 Youth Scientists of Shandong Province of China (2008BS01011), National Cryptologic Development Foundation of China, and the Youth Research Foundation of Qingdao University.

REFERENCES

[1] R. Anderson, “Two remarks on public key cryptology,” Invited Lecture, In: the 4th ACM Conference on Computer and Communications Security, 1997.

[2] M. Bellare, S. Miner, “A forward-secure digital signature scheme,” In: Wiener, M.J. (ed.) Advances in Cryptology-CRYPTO’99. LNCS, vol. 1666, pp. 431–448. Springer, Heidelberg, 1999.

[3] M. Abdalla, L. Reyzin, “A new forward-secure digital signature scheme,” In: Okamoto, T. (ed.) Advances in Cryptology- ASIACRYPT 2000. LNCS, vol. 1976, pp. 116-129. Springer, Heidelberg, 2000.

[4] H. Krawczyk, “Simple forward-secure signatures for any signature scheme,” In: the 7th ACM Conference on Computer and Communications Security. pp. 108-115. ACM Press, New York, 2000.

[5] G. Itkis, L. Reyzin, “Forward-secure signatures with optimal signing and verifying,” In: Kilian, J. (ed.) Advances in Cryptology-CRYPTO 2001. LNCS, vol. 2139, pp. 499–514. Springer, Heidelberg, 2001.

[6] A. Kozlov, L. Reyzin, “Forward-secure signatures with fast key update,” In: Cimato, S., Galdi, C., Persiano, G. (Eds.) the Proc of security in communication Networks. LNCS, vol. 2576, pp. 247-262. Springer, Heidelberg, 2002.

[7] T. Maklin, D. Micciancio, S. Miner, “Efficient generic forward-secure signatures with an unbounded number of time periods,” In: Knudsen, L. (ed.) Advances in Cryptology-EUROCRYPT 2002. LNCS, vol. 2332, pp. 400-417. Springer, Heidelberg, 2002.

[8] C. Gentry, A. Silverberg, “Hierarchical ID-based cryptography,” In: Zheng, Y. (ed.) Advances in Cryptology-Asiacrypt 2002. LNCS, vol. 2501, pp. 548-566. Springer, Heidelberg, 2002.

[9] F. Hu, C. H. Wu, J. D. Irwin, “A new forward-secure signature scheme using bilinear maps,” Cryptology ePrint Archive, Report 2003/188, 2003.

[10] B.G. Kang, J. H. Park, S.G. Halm, “A new forward-secure signature scheme,” Cryptology ePrint Archive, Report 2004/183, 2004.

[11] J. Yu, F. Y. Kong, X. G. Cheng, R. Hao, G. W. Li, “Construction of Yet Another Forward-secure Signature Scheme Using Bilinear Maps,” In: the second international conference on provable security (ProvSec 2008). LNCS, vol. 5324, pp. 83-97. Springer, Heidelberg, 2008.

[12] X. Boyen, H. Shacham, E. Shen, B. Waters, “Forward-secure Signatures with Untrusted Update,” In: the 13th ACM conference on Computer and communications security. pp. 191-200. ACM Press, New York, 2006.

[13] B. Libert, J. Jacques, M. Yung, “Forward-Secure Signatures in Untrusted Update Environments: Efficient and Generic Constructions,” In: the 14th ACM conference on Computer and communications security. pp. 266-275. ACM Press, New York, 2007.

[14] M. Bellare, B. Yee, “Forward-security in private-key cryptography,” In: Joye, M. (Ed.) Topics in Cryptology-CT-RSA 2003. LNCS, vol. 2612, pp. 1-18. Springer, Heidelberg, 2003.

[15] R. Canetti, S. Halevi, J. Katz, “A forward-secure public-key encryption scheme,” In: Biham, E. (ed.) Advances in Cryptology-EUROCRYPT 2003. LNCS, vol. 2656, pp. 255-271. Springer, Heidelberg, 2003.

[16] M. Abdalla, S. Miner, C. Namprempre, “Forward-secure threshold signature schemes,” In: Naccache, D. (ed.) Topics in Cryptology–CT-RSA 2001. LNCS, vol. 2020, pp. 441-456. Springer, Heidelberg, 2001.

[17] Z. J. Tzeng, W.G. Tzeng, “Robust forward signature schemes with proactive security,” In: Kim, K. (ed.) Public-Key Cryptography (PKC 2001). LNCS, vol. 1992, pp. 264-276. Springer, Heidelberg, 2001.

[18] H. Wang, G. Qiu, D. Feng, G. Xiao, “Cryptanalysis of Tzeng-Tzeng Forward-Secure Signature Schemes,” IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences E89-A(3), 822-825, 2006.

[19] J. Yu, F. Y. Kong, R. Hao, “Forward-secure Threshold Signature Scheme from Bilinear Pairings,” In: Wang, Y., Cheung, Y., Liu, H. (Eds.) the Second International Conference on Computational Intelligence and Security. LNAI, vol. 4456, pp. 587-597. Springer, Heidelberg, 2007.

[20] Y. Dodis, J. Katz, S. Xu, M. Yung, “Key-insulated public key cryptosystems,” In: Knudsen, L. (ed.) Advances in Cryptology- Eurocrypt 2002. LNCS, vol. 2332, pp. 65-82. Springer, Heidelberg, 2002.

[21] Y. Dodis, J. Katz, S. Xu, M. Yung, “Strong key-insulated signature scheme,” In: Desmedt, Y. (ed.) Advances in

Public key Cryptography-PKC 2003. LNCS, vol. 2567, pp. 130-144. Springer, Heidelberg, 2003.
[22] Y. Zhou, Z. Cao, Z. Chai, “Identity Based Key Insulated Signature,” In: Chen, K., Deng, R., Lai, X., Zhou, J. (Eds.) the Second International Conference Information Security Practice and Experience (ISPEC 2006). LNCS, vol. 3903, pp. 226-234. Springer, Heidelberg, 2006.
[23] B. Libert, J. Quisquater, M. Yung, “Parallel Key-Insulated Public Key Encryption Without Random Oracles,” In: Okamoto, T., Wang, X. (Eds.) Advances in Public Key Cryptography–PKC 2007. LNCS, vol. 4450, pp. 298-314. Springer, Heidelberg, 2007.
[24] G. Itkis, L. Reyzin, “SiBIR: Signer-base intrusion-resilient signatures,” In: Yung, M. (ed.) Advances in Cryptology-CRYPTO 2002. LNCS, vol. 2442, pp. 499-514. Springer, Heidelberg, 2002.
[25] Y. Dodis, M. Franklin, J. Katz, A. Miyaji, M. Yung, “Intrusion resilient public-key encryption,” In: Joye, M. (ed.) Topics in Cryptology–CT-RSA 2003. LNCS, vol. 2612, pp. 19-32. Springer, Heidelberg, 2003.
[26] Y.Dodis, M. Franklin, J. Katz, A. Miyaji, M. Yung, “A generic construction for intrusion-resilient public-key encryption,” In: Okamoto, T. (ed.) Topics in Cryptology–CT-RSA 2004. LNCS, vol. 2964, pp. 81-98. Springer, Heidelberg, 2004.
[27] G. Itkis, “Intrusion-resilient signature: Generic constructions, or Defeating a strong adversary with minimal assumption,” In: Cimato, S., Galdi, C., Persiano, G. (Eds.) Security in communication Networks. LNCS, vol. 2576, pp. 102-118. Springer, Heidelberg, 2002.
[28] NEC Research & Development 71 (1983) 1－8. David Chaum: Blind Signatures for untraceable payments; Crypto’82, pp. 199－203, Plenum Press, New York (1983).
[29] K. Ohta, T. Okamoto, Multi-signature scheme secure against active insider attacks, IEICE Transactions on Fundamentals of Electronics Communications and Computer Sciences E82-A(1) (1999) 21－31.
[30] S. M. C. Sherman, C. K. H. Lucas, S.M. Yiu., K.P Chow, “Forward-secure multisignature and blind signature schemes,” Applied Mathematics and Computation, Elsevier, 168(2005), 895-908.
[31] C. David, Blind Signatures for untraceable payments; Crypto’82, pp.199－203, Plenum Press, New York, 1983.
[32] A. Boldyreva. Efficient threshold signature, multisignature, and blind signature schemes based on the Gap-Diffie-Hellman-group signature scheme. In PKC’03, LNCS, vol. 567, pp. 31－46. Springer, Heidelberg, 2003.
[33] F. Zhang and K. Kim. Efficient ID-based blind signature and proxy signature from bilinear pairings. In Proc. ACISP’03, LNCS, vol. 2727, pp. 312－323. Springer, Heidelberg, 2003.
[34] F. Zhang, R. Safavi-Naini, and W. Susilo. Efficient verifiably encrypted signature and partially blind signature from bilinear pairings. In Proc. INDOCRYPT03, LNCS, vol. 2904, pp. 191-204. Heidelberg, 2003.
[35] S. H.Wang, F.Bao and R. H. Deng, Cryptanalysis of a Forward-secure Blind Signature Scheme with Provable Security Information and Communications Security 2005, LNCS Vol. 3783, pp. 53-60, Heidelberg, 2005.
[36] R. Gennaro, S. Jarecki, H. Krawczyk and T. Rabin. Secure distributed key generation for discrete-log based cryptosystems, Advances in Cryptology-Eurocrypt’99. LNCS Vol. 1592, pp. 295-310, Heidelberg, 1999.

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.

Fanyu Kong was born in China in 1978. 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 of computer science in the institute of Network Security at Shandong University, China, in 2006. He is currently a fellow in the institute of Network Security at Shandong University, China. His research interests include cryptography and network security.

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

Xiangguo Cheng was born in China in 1969. He received PhD degrees from Xian dianzi University, China, in 2006.

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.

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

Rong Hao was born in China in 1976. He received the BS, MS degrees in computer science from Jinan University and 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.

Yangkui Chen was born in China in 1984. He received the BS degrees in computer science from Shandong University of Technique. He is currently a postgraduate at Qingdao University. His research interests include cryptography and network security.

Xuliang Li was born in China in 1977. He is currently a researcher of computer science at Qingdao University, China. His research interests include digital signature, cryptographic protocol.

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, 2003, and 2007, respectively. He is currently a lecturer of computer science at Shandong Jianzhu University, China, in 2006. His research interests include encryption, digital signature, cryptographic protocol.