An Improved Online/Offline Identity-based Signature Scheme for WSNs

Ya Gao[1], Peng Zeng[1], Kim-Kwang Raymond Choo[2], and Fu Song[1]
(Corresponding author: Peng Zeng)

Shanghai Key Laboratory of Trustworthy Computing, East China Normal University, Shanghai, China[1]
Information Assurance Research Lab, University of South Australia, Adelaide SA, Australia[2]
(Email: pzeng@sei.ecnu.edu.cn)
(Received Aug. 9, 2015; revised and accepted Jan. 23, 2016)

Abstract

Online/offline signature schemes allow the signer to generate an online signature in real-time from a precomputed offline signature when presented with a document. Such schemes are particularly useful in resource-constrained wireless sensor network applications. In this paper, we describe an identity-based online/offline signature scheme based on bilinear maps, and prove the security of the scheme assuming the intractability of the Computational Diffie-Hellman Problem. More precisely, under the random oracle model, our scheme is proved to be secure against existential forgery on adaptively chosen message attack. As an extension to our scheme, we demonstrate how the scheme can be extended to allow a single user to sign multiple messages.

Keywords: Bilinear pairing, identity-based signature, online/offline signature, wireless sensor network

1 Introduction

With advances in sensor technologies in recent times, wireless sensor networks (WSNs) are increasingly popular in commercial, government and military settings (see [17, 24, 26, 31]). A WSN is a network of spatially distributed autonomous sensors deployed to monitor physical or environmental conditions, such as temperature and pressure. Sensor nodes cooperatively pass their data through the network to a main location. A WSN environment typically consists of a large number of resource-constrained sensor nodes and several control nodes (also known as base stations) [18]. Similar to Mobile Ad Hoc Networks [1, 2], the open nature of wireless communication result in WSNs being vulnerable to a wider range of attacks. Therefore, providing authentication for sensor data is of utmost importance in WSN applications [16, 27, 38, 39].

Since sensor nodes are typically resource constrained (e.g. in terms of memory and battery power), symmetric-key-based μTESLA-like schemes [12, 22, 23, 28] are more appropriate for actual deployment on the nodes due to their energy efficiency. However, these schemes are vulnerable to energy-depleting denial of service (DoS) attacks [3, 21]. Secret key distribution problem between senders and receivers is also a challenge when deploying WSNs [34]. In the last few years, several schemes based on public key cryptography [5, 9, 10, 14] have been proposed to provide real-time authentication and eliminate the key distribution/management problem, which reduces the protocol overhead. In a traditional public key infrastructure deployment, we would require a trusted certification authority to issue a certificate in order to authenticate the user's public key [11]. However, such an approach consumes substantial bandwidth and power due to the need for transmitting and verification of public key certificates [33, 34].

Shamir [32] introduced identity-based (ID-based) cryptosystems and signature schemes, which eliminate the need for checking the validity of certificates. A user can use his name, e-mail address or other identity attributes as the public key, and therefore, ID-based cryptography is a viable option for WSNs. For example, when a new node joins the network, other nodes do not need to keep the certificate in order to communicate in a secure and authenticated way. In order to further reduce the computational overhead of signature generation, online/offline technology is deployed in WSNs. An online/offline signature scheme was introduced by Even et al. [13], where the signing of a message is separated into two phases. The first phase is performed offline, which can be executed before the message to be signed is known. Upon receiving the message to be signed, the second phase is performed online, which utilizes the precomputation of the first phase. Activities that require significant computation resources, such as exponentiation, should be avoided in the online phase for efficiency. This property is useful in WSNs. The offline phase can be performed by the powerful base station, while the online phase can be executed by the sensor nodes [36, 37].

The first online/offline ID-based signature scheme is, perhaps, proposed by Xu et al. [35]. In the scheme, whenever a signature needs to be generated, the signer will execute the offline phase. In a WSN, when the offline phase is performed at the base station, the sensor nodes need to obtain the next offline signature from the base station whenever the node is generating a signature. This will resulted in increased communication overheads. It was subsequently discovered that Xu's scheme does not achieve the claimed security property [20]. In a separate work, Liu et al. [25] presented an online/offline ID-based signature scheme, which allows the signer to reuse the offline precomputed information in polynomial time. However, Kar [19] demonstrated that Liu et al.'s scheme does not include the case that randomly selects string contains all $0s$ or all $1s$ or the position of 1 in odd or even place. For these cases, the scheme would be vulnerable to malicious attacks. An improved scheme was then proposed. However, in this paper, we explain that the verification equation in Kar's scheme [19] is invalid, and it does not achieve the claimed security property (i.e. the signature is forgeable). As an illustration, we will show how to forge a signature in Kar's scheme. We also propose an improved scheme, with an accompanying security analysis.

The rest of the paper is organized as follows. In Section 2, we review the relevant definitions and outline the framework of the online/offline ID-based signature scheme. In Section 3, we revisit Kar's scheme and reveal a previously unpublished vulnerability by demonstrating how a signature can be forged. Our scheme is described in Section 4, followed by the security proof. We extend the basic scheme to an aggregate signature scheme in Section 5, before concluding the paper in Section 6.

2 Preliminaries

2.1 Bilinear Pairings

Definition 1. *Let k be a security parameter and q be a k-bit prime number. Let $\mathbb{G}_1$ denote a cyclic additive group of prime order q and $\mathbb{G}_2$ a cyclic multiplicative group with the same order. We assume that the discrete logarithm problem is hard in both $\mathbb{G}_1$ and $\mathbb{G}_2$. Let $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$ be a map with the following properties:*

1) *Bilinearity: For any $P, Q \in \mathbb{G}_1$ and $a, b \in \mathbb{Z}_q^*$, $\hat{e}(aP, bQ) = \hat{e}(P, Q)^{ab}$.*
2) *Non-degeneracy: There exists $P, Q \in \mathbb{G}_1$ such that $\hat{e}(P, Q) \neq 1_{\mathbb{G}_2}$. Therefore, when P is a generator of $\mathbb{G}_1$, $\hat{e}(P, P)$ is a generator of $\mathbb{G}_2$.*
3) *Computability: There is an efficient algorithm to compute $\hat{e}(P, Q)$ for all $P, Q \in \mathbb{G}_1$.*

The above bilinear map is also known as a bilinear pairing. The map $\hat{e}$ can be derived from either Weil pairing or Tate pairing on an elliptic curve over a finite field, and we refer the reader to [4, 6, 8, 15] for a more comprehensive description.

2.2 Mathematical Assumption

Let $\mathbb{G}$ be an abelian group of prime order q and P a generator of $\mathbb{G}$. We describe the following three mathematical problems in the additive group $(\mathbb{G}, +)$.

Discrete Logarithm Problem (DLP): Given $P, Q \in \mathbb{G}$, find an integer $n \in \mathbb{Z}_q^*$, such that $Q = nP$ whenever such an integer exists.

Decision Diffie-Hellman Problem (DDHP): For $a, b, c \in \mathbb{Z}_q^*$, given $P, aP, bP, cP \in \mathbb{G}$, decide whether $c \equiv ab \mod q$.

Computational Diffie-Hellman Problem (CDHP): For $a, b \in \mathbb{Z}_q^*$, given $P, aP, bP \in \mathbb{G}$, compute abP.

CDHP assumption: There exists no algorithm running in polynomial time, which can solve the CDHP problem with non-negligible probability.

(t', ε')-CDH group: A probabilistic algorithm $\mathcal{A}$ is said (t', ε')-break the CDHP in $\mathbb{G}$ if $\mathcal{A}$ runs at most time t', computes the CDHP with an advantage of at least ε'. We say that $\mathbb{G}$ is a (t', ε')-CDH group if no probabilistic algorithm $\mathcal{A}$ (t', ε')-breaks the CDHP in $\mathbb{G}$.

2.3 Framework

Definition 2. *The online/offline ID-based signature scheme comprises five polynomial time algorithms, namely: Setup, Extract, Offline Sign, Online Sign, Verify.*

Setup. *The master key and parameter generation algorithm is a probabilistic algorithm. On input a security parameter 1^k, the algorithm will output a master key msk and a parameter list $params$.*

Extract. *The signing key issuing algorithm is a deterministic algorithm. On input a user's identity id and a master key msk, the algorithm will return a pair of matching public and secret keys (pk_{id}, sk_{id}).*

Offline Sign. *The offline signing algorithm is a probabilistic algorithm. On input a parameter list $params$, the algorithm will return the generated offline signature σ_{off}.*

Online Sign. *The online signing algorithm is a probabilistic algorithm. On input a parameter list $params$, an identity id, a message m, and an offline signature σ_{off}, the algorithm will return the generated signature σ.*

Verify. *The verification algorithm is a deterministic algorithm. On input a parameter list $params$, an identity id, a message m, and a signature σ, the algorithm will return $'accept'$ if σ is valid and $'reject'$ otherwise.*

3 Revisiting Kar's Online/Offline ID-based Signature Scheme

3.1 Kar's Scheme

Kar's Scheme [19] comprises the following five polynomial time algorithms.

Setup. Given security parameters k, the Private Key Generator (PKG) chooses two groups $\mathbb{G}_1$ and $\mathbb{G}_2$ both of prime order q, a generator P of $\mathbb{G}_1$, a bilinear map $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$ and two collision resistant hash functions $H_1 : \{0,1\}^* \to \mathbb{G}_1$, $H_2 : \{0,1\}^* \to \mathbb{Z}_q^*$. Next, PKG will choose a master-key $s \in \mathbb{Z}_q^*$ and compute $P_{pub} = sP$. The system public parameters are given by $\mathcal{P} = (\mathbb{G}_1, \mathbb{G}_2, q, \hat{e}, P, P_{pub}, H_1, H_2)$.

Extract. Given an identity $ID \in \{0,1\}^*$, the secret key will be $d_{ID} = s \cdot Q_{ID}$, where $Q_{ID} = H_1(ID)$.

Offline Sign. At the offline stage, the signer computes

$$\hat{\alpha}_i = \hat{e}(P, P_{pub})^{2^i}, \ \forall \ i = 0, 1, \ldots, |q| - 1.$$

Online Sign. During this phase, the signer randomly selects $\beta \in \mathbb{Z}_q^*$ and computes two index sets $\mathcal{D} = \{1 \leq i \leq |q| \mid \beta[i] = 1\}$ and $\mathcal{C} = \{1 \leq i \leq |q| \mid \beta[i] = 0\}$, where $\beta[i]$ is the i^{th} bit of β. Next, the signer will compute $\psi_1 = \prod_{i \in \mathcal{D}} \hat{\alpha}_{i-1}$, $\psi_2 = \prod_{i \in \mathcal{C}} \hat{\alpha}_{i-1}$ and $\alpha = \psi_1 \psi_2$. Then, the signer randomly selects $\gamma \in \mathbb{Z}_q^*$ and computes $U = \gamma \cdot P$, $r = H_2(ID, U||m)$, and $V = (\gamma + \beta) \cdot P_{pub} + r d_{ID}$. The signature is $\sigma = (\alpha, U, V)$.

Verify. The signature is valid only if the following equation holds:

$$\hat{e}(V, P) \stackrel{?}{=} \alpha \cdot \hat{e}(Q_{ID}, P_{pub})^r \cdot \hat{e}(U, P_{pub}). \quad (1)$$

3.2 Previously Unpublished Vulnerabilities

We will now show that Equation (1) does not hold for general cases, even in the event that (α, U, V) is a valid signature for the message m and the identity ID. First we have

$$\begin{aligned}
\hat{e}(V, P) &= \hat{e}((\gamma + \beta)P_{pub} + r d_{ID}, P) \\
&= \hat{e}((\gamma + \beta)P_{pub}, P) \cdot \hat{e}(r d_{ID}, P) \\
&= \hat{e}(P_{pub}, (\gamma + \beta)P) \cdot \hat{e}(rsQ_{ID}, P) \\
&= \hat{e}(P_{pub}, \gamma P) \cdot \hat{e}(P_{pub}, \beta P) \cdot \hat{e}(rQ_{ID}, sP) \\
&= \hat{e}(P_{pub}, U) \cdot \hat{e}(P_{pub}, P)^{\beta} \cdot \hat{e}(rQ_{ID}, P_{pub}) \\
&= \hat{e}(P_{pub}, P)^{\beta} \cdot \hat{e}(Q_{ID}, P_{pub})^r \cdot \hat{e}(U, P_{pub}).
\end{aligned}$$

Thus, Equation (1) holds if, and only if, $\alpha = \hat{e}(P_{pub}, P)^{\beta}$. However, we have

$$\begin{aligned}
\alpha &= \psi_1 \psi_2 \\
&= (\prod_{i \in \mathcal{D}} \hat{\alpha}_{i-1})(\prod_{i \in \mathcal{C}} \hat{\alpha}_{i-1}) \\
&= \hat{\alpha}_0 \hat{\alpha}_1 \cdots \hat{\alpha}_{|q|-1} \\
&= \hat{e}(P, P_{pub})^{2^0} \hat{e}(P, P_{pub})^{2^1} \cdots \hat{e}(P, P_{pub})^{2^{|q|-1}} \\
&= \hat{e}(P, P_{pub})^{2^0 + 2^1 + \cdots + 2^{|q|-1}} \\
&= \hat{e}(P, P_{pub})^{2^{|q|} - 1}.
\end{aligned}$$

Since β is randomly selected from $\mathbb{Z}_q^*$, it is clear that $\beta \neq 2^{|q|} - 1 \mod q$ in general, which results in

$$\alpha \neq \hat{e}(P_{pub}, P)^{\beta};$$

thus, Equation (1) does not hold.

In addition to the above design flaw, we will show that the scheme is vulnerable to an existential forgery attack, in violation of their security claim. We reasonably assume that $\mathcal{A}$ is an attacker who has the public parameters

$$\mathcal{P} = (\mathbb{G}_1, \mathbb{G}_2, q, \hat{e}, P, P_{pub}, H_1, H_2).$$

$\mathcal{A}$ can execute the following steps to forge a signature $\sigma' = (\alpha', U', V')$ for a message m' and a legitimate identity ID.

1) $\mathcal{A}$ selects $U' \in \mathbb{G}_1$ and computes
$$r' = H_2(ID, U'||m').$$
2) $\mathcal{A}$ selects $V' \in \mathbb{G}_1$ and computes
$$\alpha' = \hat{e}(V', P) \cdot \hat{e}(r' Q_{ID} + U', P_{pub})^{q-1},$$
where $Q_{ID} = H_1(ID)$.
3) $\mathcal{A}$ sends the forgery signature $\sigma' = (\alpha', U', V')$ for the message m' and the identity ID to the verifier.

When the verifier receives the signature $\sigma' = (\alpha', U', V')$ for the message m' and the identity ID, the verifier will compute $r' = H_2(ID, U'||m')$ and check whether Equation (2) holds.

$$\hat{e}(V', P) \stackrel{?}{=} \alpha' \cdot \hat{e}(Q_{ID}, P_{pub})^{r'} \cdot \hat{e}(U', P_{pub}) \quad (2)$$

We now obtain:

$$\begin{aligned}
& \alpha' \cdot \hat{e}(Q_{ID}, P_{pub})^{r'} \cdot \hat{e}(U', P_{pub}) \\
= \ & \alpha' \cdot \hat{e}(r' Q_{ID}, P_{pub}) \cdot \hat{e}(U', P_{pub}) \\
= \ & \alpha' \cdot \hat{e}(r' Q_{ID} + U', P_{pub}) \\
= \ & \hat{e}(V', P) \cdot \hat{e}(r' Q_{ID} + U', P_{pub})^{q-1} \cdot \hat{e}(r' Q_{ID} + U', P_{pub}) \\
= \ & \hat{e}(V', P) \cdot \hat{e}(r' Q_{ID} + U', P_{pub})^{q} \\
= \ & \hat{e}(V', P)
\end{aligned}$$

The above equation holds because the group $\mathbb{G}_2$ has prime order q. Therefore, the forgery signature $\sigma' = (\alpha', U', V')$ for the message m' and the identity ID will always be successfully verified. In other words, it is trivial to forge a signature. Consequently, this violates the claim by Kar that the scheme is secure against existential forgery on chosen message attack.

4 Our Proposed Signature Scheme

4.1 The Basic Scheme

In this section, we propose an improved online/offline signature scheme whose security is based on the assumption that CDHP is hard to solve. Our scheme contains the following five polynomial time algorithms.

Setup. Given a security parameter $k \in \mathbb{Z}$, this algorithm works as follows:

1) Generates a prime q, two groups $\mathbb{G}_1$ and $\mathbb{G}_2$ of order q and a bilinear pairing $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$. Chooses a generater P of $\mathbb{G}_1$.
2) Selects a random $s \in \mathbb{Z}_q^*$ as the master key, and sets $P_{pub} = sP$.
3) Chooses two cryptographic hash functions $H_1 : \{0,1\}^* \to \mathbb{G}_1$ and $H_2 : \{0,1\}^* \to \mathbb{Z}_q^*$, which will be viewed as random oracles in our security proof. The system parameters are

$$Params = \{\mathbb{G}_1, \mathbb{G}_2, P, P_{pub}, q, \hat{e}, H_1, H_2\}.$$

Extract. For a given identity $ID \in \{0,1\}^*$, the algorithm computes the associated private key $S_{ID} = s \cdot H_1(ID)$, where $Q_{ID} = H_1(ID)$ plays the role of the associated public key.

Offline Sign. During the offline stage, the signer computes:

$$Y_i = \hat{e}(P_{pub}, P)^{2^i},\ i = 0, 1, \ldots, \ell,$$

where $\ell = |q| - 1$.

Online Sign. During the online stage, given a private key S_{ID} and a message $m \in \{0,1\}^*$, the signer computes the followings:

1) Randomly chooses a number $y \in \mathbb{Z}_q^*$ and computes
$$Y = \prod_{0 \le i \le \ell} Y_i^{y[i]},$$
where $y[i]$ denotes the i^{th} bit of y, $0 \le i \le \ell$.
2) Randomly chooses $x \in \mathbb{Z}_q^*$, and computes $R = xP$ and
$$h = H_2(m, R, Y).$$
3) Computes $Z = (x + y)P_{pub} + hS_{ID}$.

The signature is $\sigma = (Y, R, Z)$.

Verify. In order to verify the signature σ of a message m for an identity ID, the verifier computes the followings:

1) Computes $h = H_2(m, R, Y)$.
2) Verifies whether the following equation holds.

$$\hat{e}(Z, P) \stackrel{?}{=} Y \cdot \hat{e}(R + hQ_{ID}, P_{pub}) \qquad (3)$$

Accepts if the above verification returns true, and rejects otherwise.

Consistency. Let $\sigma = (Y, R, Z)$ be a valid signature of a message m for an identity ID (in the case $Z = (x + y)P_{pub} + hS_{ID}$, $R = xP$, $h = H_2(m, R, Y)$, and $Y = \prod_{0 \le i \le \ell} Y_i^{y[i]}$), we have

$$\begin{aligned}
\hat{e}(Z, P) &= \hat{e}((x + y)P_{pub} + hS_{ID}, P) \\
&= \hat{e}((x + y)P_{pub}, P) \cdot \hat{e}(hS_{ID}, P) \\
&= \hat{e}(P_{pub}, (x + y)P) \cdot \hat{e}(hsQ_{ID}, P) \\
&= \hat{e}(P_{pub}, xP) \cdot \hat{e}(P_{pub}, yP) \cdot \hat{e}(hQ_{ID}, sP) \\
&= \hat{e}(P_{pub}, R) \cdot \hat{e}(P_{pub}, P)^y \cdot \hat{e}(hQ_{ID}, P_{pub}) \\
&= \hat{e}(P_{pub}, P)^y \cdot \hat{e}(R + hQ_{ID}, P_{pub})
\end{aligned}$$

Then, Equation (3) holds if and only if $Y = \hat{e}(P_{pub}, P)^y$. On the other hand,

$$\begin{aligned}
Y &= \prod_{0 \le i \le \ell} Y_i^{y[i]} \\
&= Y_0^{y[0]} Y_1^{y[1]} \ldots Y_\ell^{y[\ell]} \\
&= \hat{e}(P_{pub}, P)^{y[0]2^0} \cdot \hat{e}(P_{pub}, P)^{y[1]2^1} \cdots \hat{e}(P_{pub}, P)^{y[\ell]2^\ell} \\
&= \hat{e}(P_{pub}, P)^{y[0]2^0 + y[1]2^1 + \cdots + y[\ell]2^\ell} \\
&= \hat{e}(P_{pub}, P)^y.
\end{aligned}$$

Thus, we show the consistency of our signature scheme. In the next section, we will prove that our signature scheme is secure against existential forgery on adaptively chosen message attack under the CDHP assumption.

4.2 Security Proof

Let $\mathcal{S} = (\textbf{Setup}, \textbf{Extract}, \textbf{Offline Sign}, \textbf{Online Sign}, \textbf{Verify})$ denotes an online/offline ID-based signature scheme. We consider the following game, denoted by $\text{Game}_{\mathcal{S},\mathcal{A}}^{\text{EUF-ACM}}$, involving a probabilistic polynomial time algorithm $\mathcal{A}$:

1) The challenger, denoted by $\mathcal{F}$, runs the **Setup** algorithm to generate the system parameters *Params* and sends them to $\mathcal{A}$.
2) $\mathcal{A}$ performs the following queries as he wants:
   - Hash function query. $\mathcal{F}$ computes the value of the hash function for the requested input and sends the value to $\mathcal{A}$.
   - Extract query. When $\mathcal{A}$ produces an identity id, $\mathcal{F}$ will return the private key sk_{id} corresponding to id, which is obtained by running **Extract**.

- Sign query. Proceeding adaptively, $\mathcal{A}$ requests signatures on at most q_s messages of his choice $m_1, \ldots, m_{q_s} \in \{0,1\}^*$. $\mathcal{F}$ responds to each query with a signature σ_i $(1 \leq i \leq q_s)$, which is obtained by running **Offline Sign** and **Online Sign**.

3) After a polynomial number of queries, the adversary $\mathcal{A}$ produces a tuple (id^*, m^*, σ^*) whose secret key was not asked in any Extract queries and the pair (id^*, m^*) was not asked in any Sign queries. $\mathcal{A}$ wins the game if σ^* is a valid signature of m^* for id^*.

Definition 3. *An adversary $\mathcal{A}$ $(t, q_h, q_e, q_s, \varepsilon)$-breaks an online-offline ID-based signature scheme $\mathcal{S}$ if $\mathcal{A}$ wins the game $Game^{EUF\text{-}ACM}_{\mathcal{S},\mathcal{A}}$ with a non-negligible advantage (i.e. advantage of at least ε), running time at most t, and Hash functions, Extract and Sign queries at most q_h, q_e, q_s times, respectively. $\mathcal{S}$ is considered $(t, q_h, q_e, q_s, \varepsilon)$-existentially unforgeable under adaptively chosen message attacks if no adversary $(t, q_h, q_e, q_s, \varepsilon)$-breaks $\mathcal{S}$.*

We now prove the following lemma using the technique used in the BLS scheme [8].

Lemma 1. *Let $\mathbb{G}_1$ be an additive group and $\mathbb{G}_2$ a multiplicative group, which are two (t', ε')-CDH cyclic groups of the same prime order q. Let $\hat{e}$ be a computable bilinear pairing $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$. In the random oracle model, the proposed signature scheme is $(t, q_h, q_e, q_s, \varepsilon)$-secure against existential forgery under an adaptive chosen-message attack, in which t and ε satisfy*

$$\varepsilon \geq e(1+q_e)\varepsilon', \quad t \leq t' - (q_h + 2q_e + 2q_s)t_m.$$

Here, we denote by e the base of the natural logarithm and t_m the time for computing scalar multiplication. Let q_h, q_e, q_s respectively denote the number of H_1 queries, extract query and sign query, which the adversary is allowed to make.

Proof. Suppose that $\mathcal{A}$ is a forgery algorithm who $(t, q_h, q_e, q_s, \varepsilon)$-breaks the signature scheme and outputs a valid forged signature. The algorithm $\mathcal{B}$ simulates the challenger and interacts with the forgery algorithm $\mathcal{A}$. We can then use $\mathcal{A}$ to construct a t'-time algorithm $\mathcal{B}$ and solve the CDH problem with probability of at least ε'. Let P be a generator of $\mathbb{G}_1$. We now describe algorithm $\mathcal{B}$, which computes $abP \in \mathbb{G}_1$ for a randomly given CDH instance (P, aP, bP) where $a, b \in \mathbb{Z}_q^*$.

Setup. Algorithm $\mathcal{B}$ sets $P_{pub} = aP$ as the public key, and algorithm $\mathcal{A}$ obtains the system parameters $\{P, P_{pub}\}$ from $\mathcal{B}$.

H_1-query. Algorithm $\mathcal{A}$ is allowed to query the random oracle H_1 at any time. In order to respond to these queries, algorithm $\mathcal{B}$ maintains a list of tuples $\langle ID_j, \alpha_j, \beta_j, c_j \rangle$ denoted as L_1, which is initially empty. When $\mathcal{A}$ queries the oracle H_1 at a point $ID_i \in \{0,1\}^*$, $\mathcal{B}$ responds as follows:

1) If the query ID_i already appears on the H_1-list in a tuple $\langle ID_i, \alpha_i, \beta_i, c_i \rangle$, algorithm $\mathcal{B}$ will respond with $H_1(ID_i) = \beta_i$. Otherwise, algorithm $\mathcal{B}$ generates a random coin $c_i \in \{0,1\}$, so that $Pr[c_i = 0] = 1/(1+q_e)$.
2) Algorithm $\mathcal{B}$ picks a random $\alpha_i \in \mathbb{Z}_q^*$ and computes $\beta_i = \alpha_i b^{1-c_i} P$.
3) Algorithm $\mathcal{B}$ adds the tuple $\langle ID_i, \alpha_i, \beta_i, c_i \rangle$ to the H_1-list and responds to $\mathcal{A}$ by setting $H_1(ID_i) = \beta_i$.

Extract query. Let ID_i be an extract query issued by $\mathcal{A}$. Algorithm $\mathcal{B}$ responds to this query as follows:

1) Algorithm $\mathcal{B}$ runs H_1-query to obtain $H_1(ID_i) = \beta_i$. Let $\langle ID_i, \alpha_i, \beta_i, c_i \rangle$ be the corresponding tuple on the H_1-list. If $c_i = 0$, then algorithm $\mathcal{B}$ reports failure and terminates.
2) Otherwise, we know $c_i = 1$; hence, $\beta_i = \alpha_i P$. Algorithm $\mathcal{B}$ computes $S_{ID_i} = \alpha_i \cdot P_{pub} = a \cdot (\alpha_i P)$ and responds to algorithm $\mathcal{A}$ with S_{ID_i}.

Sign query. Let m_i be a sign query issued by $\mathcal{A}$ with the identity ID_i, algorithm $\mathcal{B}$ responds to this query as follows:

1) Algorithm $\mathcal{B}$ runs the above algorithm for responding to H_1-query to obtain a β_i such that $H_1(ID_i) = \beta_i$. Let $\langle ID_i, \alpha_i, \beta_i, c_i \rangle$ be the corresponding tuple on the H_1-list.
2) Algorithm $\mathcal{B}$ randomly picks $x_i, y_i, h_i \in \mathbb{Z}_q^*$. Then, $\mathcal{B}$ computes $Y_i = \prod_{0 \leq k < |q|} Y_k^{y_i[k]}$ (where $Y_k = \hat{e}(P_{pub}, P)^{2^k}, \forall k = 0, 1, \ldots, |q|-1$), $R_i = x_i h_i P - y_i P - \beta_i h_i$, and $Z_i = x_i h_i P_{pub}$.
3) Algorithm $\mathcal{B}$ responds to algorithm $\mathcal{A}$ with $\sigma_i = (Y_i, R_i, Z_i)$.

We also remark that σ_i is always a valid signature on the message m_i for the identity ID_i.

$$\begin{aligned} & Y \cdot \hat{e}(R + hQ_{ID}, P_{pub}) \\ = \; & \hat{e}(P_{pub}, P)^y \hat{e}(R + hQ_{ID}, P_{pub}) \\ = \; & \hat{e}(yP + R + h\beta, P_{pub}) \\ = \; & \hat{e}(xhP, P_{pub}) \\ = \; & \hat{e}(xhP_{pub}, P) \\ = \; & \hat{e}(Z, P). \end{aligned}$$

We apply the oracle replay attack (coined by Pointcheval and Stern [29, 30]). In such an attack, we need to pay attention to the problem of collisions of query results as mentioned in proof of Lemma 4 in [29]. If no collision occurs, algorithm $\mathcal{A}$ outputs a valid (ID^*, m^*, σ^*) such that the pair (ID^*, m^*) was not asked in any Sign queries. If there is no tuple on the H_1-list containing ID^*, algorithm $\mathcal{B}$ will issue such a query for $H_1(ID^*)$ to ensure that the tuple exists.

Similar to the *forking lemma* [29], by replaying $\mathcal{B}$ using the same random tape but different choices of H_1, we obtain signatures (ID, m, h, Y, R, Z) and (ID, m, h', Y, R, Z') which are valid with respect to the hash functions H_1 and H_1' with different values $h \neq h'$ on (m, Y, R), respectively.

Algorithm $\mathcal{B}$ obtains the corresponding tuple from the L_1-list. If $c = 1$, algorithm $\mathcal{B}$ outputs failure and terminates. Otherwise, we know $c = 0$; thus, $H_1(ID) = \beta = b\alpha P$. Algorithm $\mathcal{B}$ computes $Z - Z' = (h - h')S_{ID} = (h - h')sQ_{ID} = (h - h')ab\alpha P$ and $abP = (Z - Z')(h - h')^{-1}/\alpha$, where abP is the solution to the CDH instance (P, aP, bP).

We will now show that algorithm $\mathcal{B}$ solves the given CDH instance (P, aP, bP) with probability at least ε'. We analyze the three events required for algorithm $\mathcal{B}$ to succeed:

- ε_1 : Algorithm $\mathcal{B}$ does not abort as a result of any Extract queries of algorithm $\mathcal{A}$.
- ε_2 : Algorithm $\mathcal{A}$ generates a valid message-signature forgery (Y, R, Z).
- ε_3 : The event ε_2 occurs and $c = 0$ for tuples containing ID on the L_1-list.

Algorithm $\mathcal{B}$ succeeds if all these events happen, and the corresponding probability is

$$Pr[\varepsilon_1 \wedge \varepsilon_3] = Pr[\varepsilon_1] \cdot Pr[\varepsilon_2|\varepsilon_1] \cdot Pr[\varepsilon_3|\varepsilon_1 \wedge \varepsilon_2] \quad (4)$$

□

Claim 1. *The probability that algorithm $\mathcal{B}$ does not abort as a result of any Extract queries asked by algorithm $\mathcal{A}$ is at least $(1 - 1/(1 + q_e))^{q_e}$.*

Proof. We assume that $\mathcal{A}$ does not query the signature of the same message twice. The probability that algorithm $\mathcal{B}$ does not abort is at least $(1 - 1/(1 + q_e))^i$ after i $(0 \leq i \leq q_e)$ signature queries were asked by algorithm $\mathcal{A}$. It is clear that the claim is true when $i = 0$. Let ID_i be the i-th extract query asked by $\mathcal{A}$, and $\langle ID_i, \alpha_i, \beta_i, c_i \rangle$ be the corresponding tuple on the H_1-list. Before issuing the extract query, only $H_1(ID_i) = \beta_i$ depends on the random coin c_i, and distribution on $H_1(ID_i)$ is the same as c_i's. Thus, the probability that the Extract query causes $\mathcal{B}$ to abort is at most $1/(1 + q_e)$. Based on the inductive hypothesis and the independence of c_i, the probability that $\mathcal{B}$ does not abort after this signature query is at least $(1 - 1/(1 + q_e))^i$. This proves the claim; as $\mathcal{A}$ makes at most q_e extract queries, the probability that $\mathcal{B}$ does not abort is at least $(1 - 1/(1 + q_e))^{q_e} \geq 1/e$. □

Claim 2. *If $\mathcal{B}$ does not abort as a result of any extract queries of algorithm $\mathcal{A}$, then $\mathcal{A}$'s view is identical to its view in the real attack. Hence, $Pr[\varepsilon_2|\varepsilon_1] \geq \varepsilon$.*

Proof. As h_1 and h_2 are two collision resistant hash functions, responses to h_1-queries and h_2-queries are similar to those in a real attack. All responds to the Extract queries and signature queries are valid. Therefore, $\mathcal{A}$ generates a valid message-signature pair with probability of at least ε. Hence, $Pr[\varepsilon_2|\varepsilon_1] \geq \varepsilon$. □

Claim 3. *The probability that algorithm $\mathcal{B}$ does not abort after $\mathcal{A}$ outputs a valid forgery is at least $1/(1 + q_e)$. Hence, $Pr[\varepsilon_3|\varepsilon_1 \wedge \varepsilon_2] = 1/(1 + q_e)$.*

Proof. Suppose that events ε_1 and ε_2 occurred, algorithm $\mathcal{B}$ will abort only when $\mathcal{A}$ outputs a forgery message-signature pair (m, σ) and $c = 0$ in the tuple $\langle ID, \alpha, \beta, c \rangle$ on the h_1-list. If $\mathcal{A}$ did not issue an Extract query for m_i, only $H_1(ID_i)$ depends on the random coin c_i, and distribution on $H_1(ID_i)$ is the same as c_i's. Due to that $\mathcal{A}$ could not issue an extract query for m, c is independent of $\mathcal{A}$'s current view. Therefore, $Pr[c = 0|\varepsilon_1 \wedge \varepsilon_2] = 1/(1 + q_e)$. □

According to Equation (4) and using the bounds from the above claims, algorithm $\mathcal{B}$ succeeds with probability at least $1/e \cdot \varepsilon \cdot 1/(1 + q_e)$.

If algorithm $\mathcal{A}$ takes time t to run, algorithm $\mathcal{B}$ takes time t and with the time required to respond to $(q_h + q_e + q_s)$ H_1-queries, q_e extract queries, and q_s signature queries. Each hash query and extract query require one scalar multiplication in $\mathbb{G}_1$, and each signature query requires four scalar multiplications in $\mathbb{G}_1$. We assume that one scalar multiplication in $\mathbb{G}_1$ takes time t_m. Thus, algorithm $\mathcal{B}$ takes time of at most $t + (q_h + 2q_e + 2q_s)t_m$.

Lemma 2. *(Lemma 4 in [29]) If there is an algorithm $\mathcal{A}_0$ for an adaptively chosen message attack against our scheme which queries H_1, Extract, and Sign at most q_h, q_e, q_s times respectively, and has running time t_0 and advantage $\varepsilon_0 \geq 10(1 + q_s)(q_h + q_s)/2^k$, then CDHP can be solved with probability $\varepsilon' \geq 1/9$ with run time of $t' \leq 23q_h t_0/\varepsilon_0$.*

Combining the above lemmas, we have the following theorem.

Theorem 1. *If there is an algorithm $\mathcal{A}$ for an adaptively chosen message attack to our scheme which queries H_1, Extract, and Sign at most q_h, q_e, q_s times respectively, and has running time t and advantage $\varepsilon \geq 10e(1 + q_e)(1 + q_s)(q_h + q_s)/2^k$, then CDHP can be solved with probability $\varepsilon' \geq 1/9$ within running time $t' \leq 23q_h(t + (q_h + 2q_e + 2q_s)t_m)/(e(1 + q_e))$.*

5 The Extended (Aggregation) Scheme

If a sensor node is able to sign multiple messages (for example n messages) and the size of the resulting signature is smaller than n times the size of a single signature, such an aggregated signature is practical for WSN deployment due to the reduced communication overheads. We propose the following aggregation signature as an extension to our online/offline signature scheme.

Setup. Given a security parameter $k \in \mathbb{Z}$, this algorithm works as follows:

1) Generates a prime q, two groups $\mathbb{G}_1$ and $\mathbb{G}_2$ of order q and a bilinear pairing $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$. Chooses a generater P in $\mathbb{G}_1$.
2) Selects a random $s \in \mathbb{Z}_q^*$ as the master key, and sets $P_{pub} = sP$.
3) Chooses two collision resistant hash function $H_1 : \{0,1\}^* \to \mathbb{G}_1$ and $H_2 : \{0,1\}^* \to \mathbb{Z}_q^*$. The system parameters are

$$Params = \{\mathbb{G}_1, \mathbb{G}_2, P, P_{pub}, q, \hat{e}, H_1, H_2\}.$$

Extract. For a given identity $ID \in \{0,1\}^*$, compute $Q_{ID} = H_1(ID)$ and set $S_{ID} = sQ_{ID}$ as a private key of ID.

Offline Sign. At the offline stage, the signer computes:

$$Y_i = \hat{e}(P_{pub}, P)^{2^i},\ i = 0, 1, \ldots, \ell,$$

where $\ell = |q| - 1$.

Online Sign. During the online stage, given a private key S_{ID} and n messages $m_j \in \{0,1\}^*$,$1 \le j \le n$, the signer computes the followings:

1) For any $1 \le j \le n$, randomly chooses $y_j \in \mathbb{Z}_q^*$ and computes

$$Y^{(j)} = \prod_{0 \le i \le \ell} Y_i^{y_j[i]},$$

where $y_j[i]$ denotes the i^{th} bit of y_j.
2) For any $1 \le j \le n$, randomly chooses $x_j \in \mathbb{Z}_q^*$, and computes

$$h_j = H_2(m_j, R_j, Y^{(j)}),$$

where $R_j = x_j P$.
3) Computes $Z_j = (x_j + y_j)P_{pub} + h_j S_{ID},\ 1 \le j \le n$ and $Z = \sum_{j=1}^{n} Z_j$.

The signature is

$$\sigma = (Y^{(1)}, Y^{(2)}, \ldots, Y^{(n)}, R_1, R_2, \ldots, R_n, Z).$$

Verify. In order to verify the signature $\sigma = (Y^{(1)}, Y^{(2)}, \ldots, Y^{(n)}, R_1, R_2, \ldots, R_n, Z)$ for the n messages $m_j,\ j = 1, 2, \ldots, n$, and the identity ID, the verifier computes the followings:

1) Computes $h_j = H_2(m_j, R_j, Y^{(j)}),\ j = 1, 2, \ldots, n$.
2) Verifies whether the following equation holds

$$\hat{e}(Z, P) \stackrel{?}{=} \prod_{j=1}^{n} (Y^{(j)} \cdot \hat{e}(R_j + h_j Q_{ID}, P_{pub})) \quad (5)$$

Accepts if it is equal, and rejects otherwise.

Consistency. Let $\sigma = (Y^{(n)}, R_n, Z)$ be a valid signature for n messages $m_j,\ j = 1, 2, \ldots, n$, and identity ID, we have

$$\begin{aligned}
& \hat{e}(Z, P) \\
= \ & \hat{e}(\sum_{j=1}^{n} Z_j, P) \\
= \ & \hat{e}(\sum_{j=1}^{n} ((x_j + y_j)P_{pub} + h_j S_{ID}), P) \\
= \ & \prod_{j=1}^{n} \hat{e}((x_j + y_j)P_{pub} + h_j S_{ID}, P) \\
= \ & \prod_{j=1}^{n} (\hat{e}((x_j + y_j)P_{pub}, P) \cdot \hat{e}(h_j S_{ID}, P)) \\
= \ & \prod_{j=1}^{n} (\hat{e}(P_{pub}, (x_j + y_j)P) \cdot \hat{e}(h_j s Q_{ID}, P)) \\
= \ & \prod_{j=1}^{n} (\hat{e}(P_{pub}, x_j P) \cdot \hat{e}(P_{pub}, y_j P) \cdot \hat{e}(h_j Q_{ID}, sP)) \\
= \ & \prod_{j=1}^{n} (\hat{e}(P_{pub}, R_j) \cdot \hat{e}(P_{pub}, P)^{y_j} \cdot \hat{e}(h_j Q_{ID}, P_{pub})) \\
= \ & \prod_{j=1}^{n} (\hat{e}(P_{pub}, P)^{y_j} \cdot \hat{e}(R_j + h_j Q_{ID}, P_{pub}))
\end{aligned}$$

Then, Equation (5) holds if and only if

$$Y^{(j)} = \hat{e}(P_{pub}, P)^{y_j},\ j = 1, 2, \ldots, n.$$

On the other hand, for any $1 \le j \le n$, we have

$$\begin{aligned}
Y^{(j)} &= \prod_{0 \le i \le \ell} Y_i^{y_j[i]} \\
&= Y_0^{y_j[0]} Y_1^{y_j[1]} \ldots Y_\ell^{y_j[\ell]} \\
&= \hat{e}(P_{pub}, P)^{y_j[0]2^0} \ldots \hat{e}(P_{pub}, P)^{y_j[\ell]2^\ell} \\
&= \hat{e}(P_{pub}, P)^{y_j[0]2^0 + \cdots + y_j[\ell]2^\ell} \\
&= \hat{e}(P_{pub}, P)^{y_j}.
\end{aligned}$$

Thus, $\hat{e}(Z, P) = \prod_{j=1}^{n} (Y^{(j)} \cdot \hat{e}(R_j + h_j Q_{ID}, P_{pub}))$ and the verification is successful.

We refer to [7] for a detailed description of the security model and the security proof. Under the random oracle model, our aggregation signature scheme is also secure against existential forgery on adaptively chosen message attack.

6 Conclusion

In this paper, we proposed an online/offline ID-based signature scheme and proved that the scheme is secure against existential forgery on adaptively chosen message

attack in random oracle model, under the assumption that CDHP is intractable. We also extended the basic scheme to provide the ability for a user to sign multiple messages.

Acknowledgments

This work was supported in part by the NSFC-Zhejiang Joint Fund for the Integration of Industrialization and Informatization (Grant No. U1509219), the National Natural Science Foundation of China (Grant Nos. 61402179, 61321064 and 61103222), the Research Fund for the Doctoral Program of Higher Education of China (Grant No. 20110076120016), and the Pujiang Talent Project of the Shanghai Science and Technology Committee (Grant No. 14PJ1403200).

References

[1] A. Aburumman, K. K. R. Choo, "A domain-based multi-cluster SIP solution for mobile ad hoc network," in *International Conference on Security and Privacy in Communication Networks*, pp. 267–281, 2014.

[2] A. Aburumman, W. J. Seo, M. R. Islam, M. K. Khan, K. K. R. Choo, "A secure cross-domain SIP solution for mobile ad hoc network using dynamic clustering," in *International Conference on Security and Privacy in Communication Networks*, pp. 649–664,2015.

[3] A. Agah, S. K. Das, "Preventing doS attacks in wireless sensor networks: A repeated game theory approach," *International Journal of Network Security*, vol. 5, no. 2, pp. 145–153, 2007.

[4] P. S. L. M. Barreto, H. Y. Kim, B. Lynn, et al. "Efficient algorithms for pairing-based cryptosystems," in *Advances in Cryptology (Crypto'02),* Springer Berlin Heidelberg, pp. 354–369, 2002.

[5] C. Benzaid, K. Lounis, A. Al-Nemrat, et al. "Fast authentication in wireless sensor networks," *Future Generation Computer Systems,* vol. 55, pp. 362–375, 2016.

[6] D. Boneh, M. Franklin, "Identity-based encryption from the Weil pairing," in *Advances in Cryptology (Crypto'01),* Springer Berlin Heidelberg, pp. 213–229, 2001.

[7] D. Boneh, C. Gentry, B. Lynn, et al. "Aggregate and verifiably encrypted signatures from bilinear maps," in *Advances in Cryptology (Eurocrypt'03),* Springer Berlin Heidelberg, pp. 416–432, 2003.

[8] D. Boneh, B. Lynn, H. Shacham, "Short signatures from the Weil pairing," in *Advances in Cryptology (Asiacrypt'01),* Springer Berlin Heidelberg, pp. 514–532, 2001.

[9] X. Cao, W. Kou, L. Dang, et al. "IMBAS: Identity-based multi-user broadcast authentication in wireless sensor networks," *Computer Communications,* vol. 31, no. 4, pp. 659–667, 2008.

[10] C. Y. Cheng, I. C. Lin, S. Y. Huang, "An RSA-like scheme for multiuser broadcast authentication in wireless sensor networks," *International Journal of Distributed Sensor Networks,* 2015.

[11] L. Cheng, Q. Wen, Z. Jin, et al. "Cryptanalysis and improvement of a certificateless aggregate signature scheme," *Information Sciences,* vol. 295, pp. 337–346, 2015.

[12] O. Delgado-Mohatar, A. Fúster-Sabater, J. M. Sierra, "A light-weight authentication scheme for wireless sensor networks," *Ad Hoc Networks,* vol. 9, no. 5, pp. 727–735, 2011.

[13] S. Even, O. Goldreich, S. Micali "On-line/off-line digital signatures," in *Advances in Cryptology (Crypto'89) Proceedings,* Springer New York, pp. 263–275, 1990.

[14] X. Fan, G. Gong, "Accelerating signature-based broadcast authentication for wireless sensor networks," *Ad Hoc Networks,* vol. 10, no. 4, pp. 723–736, 2012.

[15] S. D. Galbraith, K. Harrison, D. Soldera, "Implementing the Tate pairing," *Algorithmic Number Theory,* Springer Berlin Heidelberg, pp. 324–337, 2002.

[16] Y. Gao, P. Zeng, K. K. R. Choo, "Multi–sender broadcast authentication in wireless sensor networks," in *IEEE Tenth International Conference on Computational Intelligence and Security,* pp. 633–637, 2014.

[17] M. Ge, K. K. R. Choo, H. Wu, Y. Yu, "Survey on key revocation mechanisms in wireless sensor networks," *Journal of Network and Computer Applications,* (In press), 2016.

[18] K. Grover, A. Lim, "A survey of broadcast authentication schemes for wireless networks," *Ad Hoc Networks,* vol. 24, pp. 288–316, 2015.

[19] J. Kar, "Provably secure online/off-line identity-based signature scheme for wireless sensor network," *International Journal of Network Security,* vol. 16, no. 1, pp. 29–39, 2014.

[20] F. Li, M. Shirase, T. Takagi, "On the security of online/offline signatures and multisignatures from acisp06," *Cryptology and Network Security,* Springer Berlin Heidelberg, pp. 108–119, 2008.

[21] W. T. Li, T. H. Feng, M. S. Hwang, "Distributed detecting node replication attacks in wireless sensor networks: A survey," *International Journal of Network Security,* VOL. 16, NO. 5, PP. 323–330, 2014.

[22] X. Li, N. Ruan, F. Wu, et al. "Efficient and enhanced broadcast authentication protocols based on multilevel μTESLA," in *IEEE International Performance Conference on Computing and Communications,* pp. 1–8, 2014.

[23] D. Liu, P. Ning, "Multilevel μTESLA: Broadcast authentication for distributed sensor networks," *ACM Transactions on Embedded Computing Systems (TECS),* vol. 3, no. 4, pp. 800–836, 2004.

[24] D. Liu, P. Ning, *Security for Wireless Sensor Networks*, Springer, 2007.

[25] J. K. Liu, J. Baek, J. Zhou, et al. "Efficient online/offline identity-based signature for wireless sensor network," *International Journal of Information Security,* vol. 9, no. 4, pp. 287–296, 2010

[26] M. Luk, A. Perrig, B. Whillock, "Seven cardinal properties of sensor network broadcast authentication," in *Proceedings of the Fourth ACM Workshop on Security of Ad Hoc and Sensor Networks,* pp. 147–156, 2006.

[27] J. Nam, K. K. R. Choo, M. Kim, J. Paik and D. Won, "Efficient and anonymous two-factor user authentication in wireless sensor networks: Achieving user anonymity with lightweight sensor computation," *PLOS ONE,* vol. 10, no. 4, pp. e0116709, 2015.

[28] A. Perrig, R. Szewczyk, J. D. Tygar, et al. "SPINS: Security protocols for sensor networks," *Wireless Networks,* vol. 8, no. 5, pp. 521–534, 2002.

[29] D. Pointcheval, J. Stern, "Security arguments for digital signatures and blind signatures," *Journal of Cryptology,* vol. 13, no. 3, pp. 361–396, 2000.

[30] D. Pointcheval, J. Stern, "Security proofs for signature schemes," in *Advances in Cryptology (Eurocrypt'96),* Springer Berlin Heidelberg, pp. 387–398, 1996.

[31] K. Ren, W. Lou, "Communication security in wireless sensor networks," *Worcester Polytechnic Institute,* 2007.

[32] A. Shamir, "Identity-based cryptosystems and signature schemes," in *Advances in Cryptology,* Springer Berlin Heidelberg, pp. 47–53, 1985.

[33] G. Sharma, S. Bala, A. K. Verma, "An improved RSA-based certificateless signature scheme for wireless sensor networks," *International Journal of Network Security,* pp. 1–8, 2014.

[34] K. A. Shim, Y. R. Lee, C. M. Park, "EIBAS: An efficient identity-based broadcast authentication scheme in wireless sensor networks," *Ad Hoc Networks,* vol. 11, no. 1, pp. 182–189, 2013.

[35] S. Xu, Y. Mu, W. Susilo, "Online/offline signatures and multisignatures for AODV and DSR routing security," *Information Security and Privacy,* Springer Berlin Heidelberg, pp. 99–110, 2006.

[36] A. C. C. Yao, Y. Zhao, "Online/offline signatures for low-power devices," *IEEE Transactions on Information Forensics and Security,* vol. 8, no. 2, pp. 283–294, 2013.

[37] A. A. Yavuz, "An efficient real-time broadcast authentication scheme for command and control messages," *IEEE Transactions on Information Forensics and Security,* vol. 9, no. 10, pp. 1733–1742, 2014.

[38] P. Zeng, Z. Cao, K. K. R. Choo, S. Wang, "Security weakness in a dynamic program update protocol for wireless sensor networks," *IEEE Communications Letters,* vol. 13, no. 6, pp. 426–428, 2009.

[39] P. Zeng, K. K. R. Choo, D. Z. Sun, "On the security of an enhanced novel access control protocol for wireless sensor networks," *IEEE Transactions on Consumer Electronics,* vol. 56, no. 2, pp. 566–569, 2010.

Ya Gao received her B.S. degree in Information Security from Shanghai University of Electric Power in 2013. She is currently pursuing the M.S. degree with the Department of Cryptography and Network Security from East China Normal University. Her research interests include cryptography, information security, and wireless sensor network.

Peng Zeng received the Ph.D. degree in computer science and technology from Shanghai Jiao Tong University in 2009 and the M.S. degree in pure mathematics from East China Normal University in 2003, respectively. He is currently an associate professor with the School of Computer Science and Software Engineering, East China Normal University, Shanghai, China. His research expertise include applied cryptography, network information security, and coding theory.

Kim-Kwang Raymond Choo received his Ph.D. degree from Queensland University of Technology in 2006. He is an associate professor at the University of South Australia. He has an interdisciplinary expertise in cyber security and digital forensics. He was named one of 10 Emerging Leaders in the Innovation category of The Weekend Australian Magazine / Microsoft's Next 100 series in 2009, and is the recipient of various awards including ESORICS 2015 Best Research Paper Award, Highly Commended Award from Australia New Zealand Policing Advisory Agency (ANZPAA), British Computer Society's Wilkes Award, Fulbright Scholarship, and 2008 Australia Day Achievement Medallion.

Fu Song is a lecturer at School of Computer Science and Software Engineering of East China Normal University, Peoples Republic of China. He received his Ph.D. in Computer Science from University Paris 7 in 2013 and M.Sc. from Software Engineering Institute of East China Normal University in 2009. His major research interests include software verification (e.g., infinite-state system modeling, temporal logics and model-checking), computer security (e.g., malware detection and binary code disassembly).