An Efficient Identity-based Online/Offline Signature Scheme without Key Escrow

Dan Liu[1], Shun Zhang[1], Hong Zhong[1], Runhua Shi[1] and Yimin Wang[1,2]
(Corresponding author: Runhua Shi)
School of Computer Science and Technology & Anhui University[1]
Hefei 230601, China
Modern Education and information Center & Anhui Agriculture University[2]
Hefei 230036, China
(Email: shirh@ahu.edu.cn)
(Received Oct. 31, 2015; revised and accepted Jan. 30 & Mar. 29, 2016)

Abstract

Recently, several identity-based signature schemes with Bilinear Pairing and Map-To-Point (MTP) functions have been introduced. However, identity-based cryptography (IBC) schemes suffer from the serious secure problem due to Key Escrow. In addition, both Bilinear Pairing and MTP function are time-consuming operations, and thus the cryptographic schemes based on these expensive operations have high computational burden. In this paper, we proposed an efficient identity-based online/offline signature scheme without using Bilinear Pairing and MTP function. Especially, the proposed scheme overcomes the key escrow problem, and achieves some good features. Furthermore, the securities of the proposed scheme were proven in the random oracle model with the hardness of elliptic curve discrete logarithm problem (ECDLP).

Keywords: Elliptic curve discrete logarithm problem, identity-based, key escrow, online/offline signature, traceability

1 Introduction

Cryptography is an important tool that enables the secure transmission of a secret message between a sender and a recipient from any potential eavesdropper. Furthermore, key management, including key generating, updating, transmitting, storing and so on, is a critical issue for most cryptosystems. Generally speaking, key management is the weakest link in the whole cryptosystem, because key leakage will directly lead to the leakage of plaintext content. Naturally, lots of cryptosystems seeks the help of Key Escrow to deal with key management [14]. Key Escrow is an arrangement in which the keys needed to decrypt the encrypted data are held in escrow so that an authorized third party may gain access to those keys under certain circumstances. For example, once the user's key is lost, according to the established rules, the user can directly get the key from the authorized third party. However, Key Escrow also brings some risks, such as key leakage or misuse, though it has low computational costs in the phase of key generation and distribution.

1.1 Related Works

In 1990, the concept of online/offline signature was first put forward by Even, Goldreich and Micali [8]. The key idea is to split the signature generation algorithm into two phases: the offline phase and the online phase. The signing algorithm executes the offline phase to perform most of the computations and stores without knowing the signed message. Once the signed message is available, the signing algorithm runs the online phase very quickly and requires only light computations. Obviously, Online/offline signatures are more useful in some power or storage limited devices, such as smart card, wireless sensor and RFID tags, because the offline phase can be executed either during device manufacturing process or as a background computation whenever the device is connected to power. Accordingly, some signature schemes can be naturally split into offline and online phases. For instance, the partial computation of some signature schemes (e.g. Schnorr, ElGamal, DSS signature schemes) does not depend on the given message, and thus it can be shifted to the offline phase directly. The first general method for transforming any ordinary signature scheme into online/offline signature scheme was proposed by Even, Goldreich and Micali [8]. Nevertheless, their method is inefficient and impractical because it increases the length of each signature by a quadratic factor. In 2001, Shamir and Thuman [21] constructed an improved online/offline signature scheme which was based on trapdoor hash function. It highly enhances the efficiency, particularly in the online phase. However, it increases the computational costs of signing and there ex-

ist trapdoor leak problems. In 2007, to overcome trapdoor leak problems, Chen [6] designed an online/offline signature scheme by utilizing the double trapdoor hash function. But, both schemes are not actually efficient or practical to be used, in generic settings.

In traditional public key cryptosystem (PKC), a random public key of one user is associated with the user by a certificate. And PKC is based on public key infrastructure (PKI), which incurs additional cost for setting up the infrastructure, and requires the public key certificate management and distribution. In addition, the generation, delivery, management, and revocation of the public key certificate need to consume huge storage space and high computational costs. To solve these problems of PKC, Shamir [20] introduced the notion of identity-based cryptography (IBC) where the public key of each user can be gained from some public information (e.g., ID number, IP address and user's name) that exclusively identifies the user and also is known to others users. In IBC, there is a trusted third party (TTP), called Private Key Generator (PKG), who computes the private key from the master secret for the users. The PKG first publishes a master public key, and retains the corresponding master private key in secret. IBC simplifies key agreement procedures of certificate-based PKI. Thenceforward, several identity-based signature schemes [7, 16, 22, 23] have been proposed, which are based on the difficulty assumption that the integer factorization problem (IFP) is hard.

The first identity-based online/offline signature scheme was introduced by Xu, Mu and Susilo [27], which combines the advantages of IBC and online/offline signature. The key certificates are eliminated, and most computations needed for the signature generation are computed before the messages are given, so that the online phase generates the signature efficiently after the messages are available. In 2006, the same authors [26] designed another efficient identity-based online/offline signature scheme for authentication in AODV routing protocol. Nevertheless, Ming et al. [28] showed that the scheme in [27] is universally forgeable. And Li et al. [15] proved that the scheme in [26] does not achieve the security. To overcome the drawback, various improved schemes have been proposed. For instance, in 2014, Kar [13] introduced a new identity-based online/offline signature scheme. Unfortunately, the scheme of Kar has high computational costs because of bilinear pairing and map-to-point (MTP) function.

1.2 Motivations and Contributions

In the identity-based signature schemes based on traditional PKI, there is a TTP, called PKG, who computes the private key from the master secret for the users. As the PKG generates and holds all secret keys for all users, a complete trust must be placed on the PKG. However, it is difficult to ensure complete trust in the real world scenario. For example, a malicious PKG can sell users' keys to get profit or even simulate any user to sign messages. This is known as the key escrow problem and it seems to be inherent in IBC. Thus, Boneh and Frankhn [5] utilized secret sharing scheme with multiple PKGs to resolve the key escrow problem, in which the master secret key is jointly computed by multiple PKGs, such that no single PKG has the knowledge of it. But this method needs an extra computation and communication overhead between multiple PKGs and the users.

In order to provide a solution, Al-Riyami and Paterson [3] introduced the notion of Certificateless Cryptography (CLC) which eliminates the use of certificates in PKC and solve the key escrow problem in IBC. The first component is chosen by the user as his/her secret value, and it is not known to PKG. On the other hand, the second component is the partial private key, which is generated by PKG. So the full private key of each user is composed with two components. In addition, CLC is not identity-based because the user has an additional random public key. Even now, some existing certificateless schemes [10, 15, 29] are showed to be insecure, while others secure schemes [9, 30] and [18, 19] have low efficiencies. Therefore it is not easy to construct a secure and efficient certificateless scheme.

Obviously, the identity-based signature scheme without key escrow can avoid some security risks. For example, it can avoid malicious PKG or adversary by obtaining user's private key to do some illegal actions. In addition, if PKG forges the secret value of a user, and binds his/her ID by more than two private keys to one adversary. That also brings security risks to the scheme.

Because of its importance and wide applicability, efficient and provably secure identity-based online/offline signature scheme is becoming the research focus. However, the most existing identity-based online/offline signature schemes use bilinear pairing and a probabilistic MTP function [13, 25, 26, 27, 28]. The relative computational cost of bilinear pairing is approximately two to three times more than the elliptic curve point multiplication. The MTP function also needs more execution time than an elliptic curve point multiplication [11, 12]. Therefore, how to design an efficient and secure identity-based online/offline signature scheme without using bilinear pairing and MTP function is still an important and practical issue.

Motivated by these concerns, in this paper, we construct an efficient identity-based online/offline signature scheme without using bilinear pairing and MTP function, which avoids the key escrow problem by adopting the idea of CLC. PKG only produces a partial private key while the user generates the other partial private key, which both them are the full private key. So the PKG does not have full knowledge of the user's private key. Further, user's public key can be directly related with his/her ID in our scheme, which is different from those schemes of CLC. It shows clearly that our scheme is more secure and efficient than some certificateless schemes [9, 18, 19]. In our scheme, even if PKG forges the signature for the user, the user can generate evidence and submit to the intercessor trusted authority (TA). According to the evidence, TA can check whether the PKG is honest. The proposed

scheme is provably secure and computationally more efficient than existing schemes. Our scheme also provides security proof under random oracle model with the difficulties of ECDLP and against adaptive chosen ID and message attacks.

The rest of this paper is organized as follows. We review some preliminary works in Section 2, and Section 3 describes the syntax of ID-based online/offline signature. In Section 4, we construct an identity-based online/offline signature scheme without key escrow problem. The security of our scheme is given in Section 5. We then give performance evaluation in Section 6. Finally, concluding remarks are given in Section 7.

2 Preliminary Works

In this section, we review certain related definitions and mathematical problem, which will be introduced in our proposed scheme.

Definition 1 (**Elliptic Curve Over** Z_p)**.** *Let $p > 3$ be an odd prime. An elliptic curve E over Z_p is defined by an equation of the from $y^2 = x^3 + ax + b$, where $x, y, a, b \in Z_p$ and $(4a^3 + 27b^2) \bmod p \neq 0$. Then we define $G_p = \{(x, y) : x, y \in Z_p and (x, y) \in E(Z_p)\} \cup \{O\}$ as the additive elliptic curve group, and the point O is known as point at infinity or zero point.*

Definition 2 (**Elliptic Curve Discrete Logarithm Problem (ECDLP))**. *Given two group elements $P \in G_1$ and $Q \in G_1$, to find an integer $a \in Z_q^*$, such that $Q = aP$ whenever such an integer exists.*

Lemma 1 (**Forking Lemma [17]**)**.** *We assume that A be a attacker within a time bound t_1, A produces a valid signature $(m, \sigma_1, h, \sigma_2)$ with probability $\varepsilon \geq 10(S+1)(S+H)/2^k$. If the triples (σ_1, h, σ_2) can be simulated without knowing the secret key, with an indistinguishable distribution probability, then there is another machine which has control over the machine obtained from A replacing interaction with the signer by simulation and produces two valid signatures $(m, \sigma_1, h, \sigma_2)$ and $(m, \sigma_1, h', \sigma_2)$ such that $h \neq h'$. We denote respectively by S and H the number of queries that A can ask to the signer and the number of queries that A can ask to the random oracle.*

3 The Syntax of ID-based Online/Offline Signature

In the section, we briefly describe the syntax of ID-based online/offline signature.

An ID-based online/offline signature scheme is composed mainly of the following five polynomial algorithms.

1) **IO_Setup:** The parameter generation is a probability algorithm running by PKG, in which it inputs a security parameter 1^k, and then outputs system parameters *params* and a master key IO_Msk.
2) **IO _Etract:** The extract algorithm is a key generation algorithm, in which it inputs the system parameters *params*, an identity ID, and a master key IO_Msk, and accordingly returns the private key IO_sk_{ID} the user.
3) **IO _OffSign:** The online signing algorithm is a probability algorithm that by inputting the system parameters *params*, and user's private key IO_sk_{ID}, the user calculates the offline signature S and stores it.
4) **IO_OnSign:** The offline signing algorithm is a probability algorithm that after inputting a message m and offline signature S, it outputs an online signature σ.
5) **IO_Verify:** The verification algorithm is a deterministic algorithm that by inputting the message m, identity ID, offline signature S, and online signature σ, finally it outputs either accept or reject.

4 The Proposed Scheme

In this section, we proposed an identity-based online/offline signature scheme without Key Escrow. Similarly, our scheme is also comprised of five polynomial time algorithms: **IO_Setup**, **IO_Extract**, **IO_OffSign**, **IO_OnSign** and **IO_Verify**. In addition, the related details of our scheme are shown in Figure 1 and Figure 2.

1) **IO_Setup:**

Let $p > 3$ be an odd prime. An elliptic curve E over Z_p is defined by an equation of the from $y^2 = x^3+ax+b$, where $x, y, a, b \in Z_p$ and $(4a^3+27b^2) \bmod p \neq 0$. Then it runs the parameter generator by inputting a security parameter $\lambda \in Z^+$ to generate a prime $q > 2^\lambda$, and choose a group G of prime order q. Suppose that P be the basic point of E, where the prime order of P is q. The group $G =< P >$ is based on the Discrete Logarithm problem. Then PKG picks a master key $s \in_R Z_q^*$ randomly and computes

$$P_{Pub} = sP. \tag{1}$$

Furthermore, PKG chooses two cryptographic hash functions $H_0 : \{0,1\}^* \rightarrow Z_q^*$ and $H_1 : \{0,1\}^* \rightarrow Z_q^*$. Finally, PKG publishes the system parameter $params = \{\lambda, q, P, G, P_{Pub}, H_0, H_1\}$ and keeps the master key s in secret.

2) **IO_Etract:**

We split this extract process into two phases: the one is to generate the partial private key; the other is to generate the secret value. Given an identity ID, PKG chooses $k \in_R Z_q^*$ randomly and computes

$$\begin{aligned} Y &= kP, \\ S_{ID} &= k + sQ_{ID} (\mathrm{mod} q). \end{aligned}$$

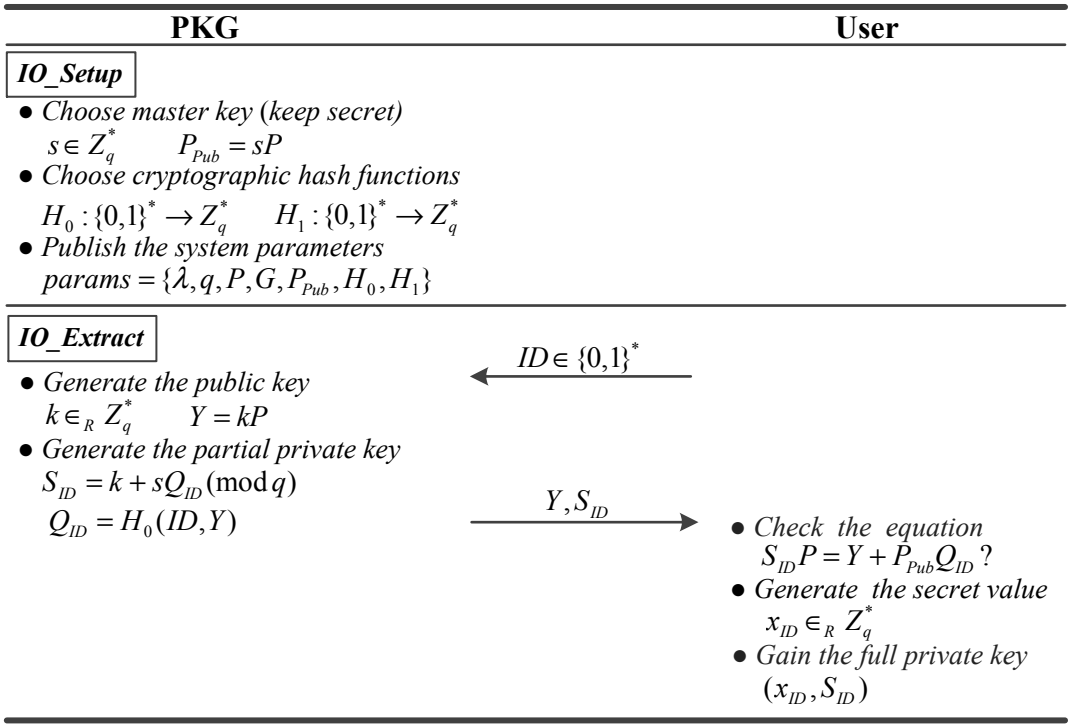

Figure 1: *IO_Setup* and *IO_Extract* phases of the proposed scheme

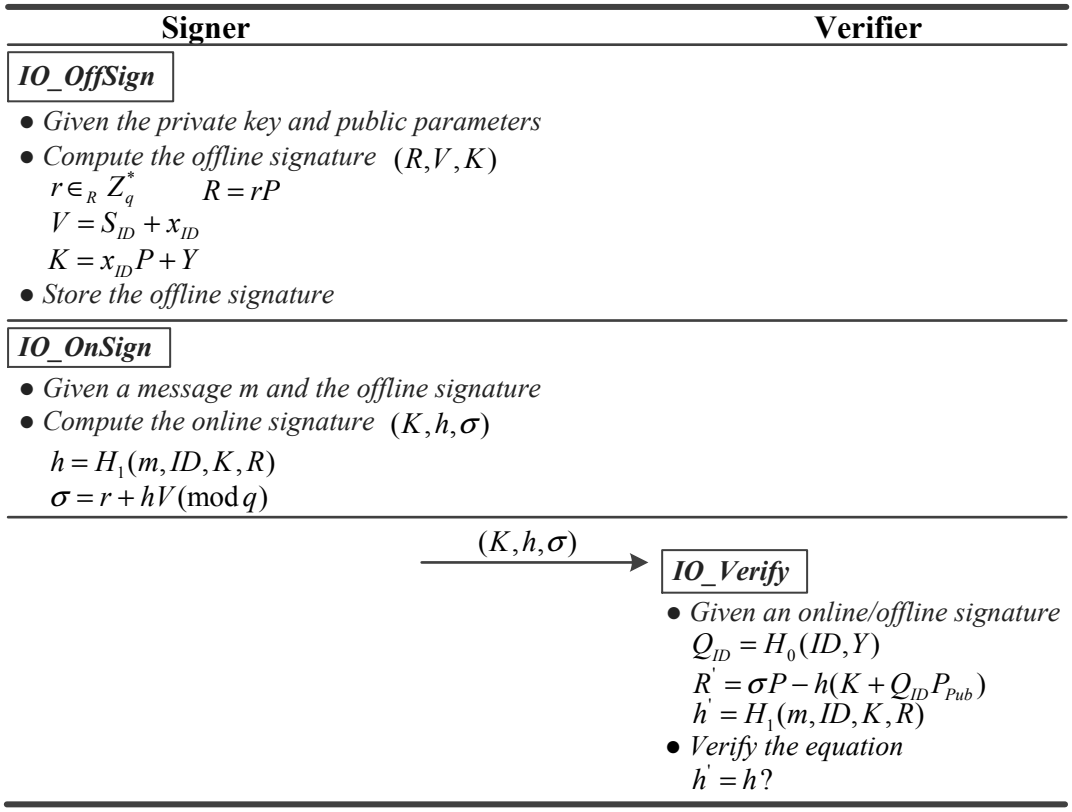

Figure 2: *IO_OffSign*, *IO_OnSign* and *IO_Verify* phases of the proposed scheme

PKG takes S_{ID} as the partial private key, and sends S_{ID} and Y to the user, where $Q_{ID} = H_0(ID, Y)$. The user may check whether the following equation holds or not

$$S_{ID}P = Y + P_{Pub}Q_{ID}. \quad (2)$$

If the equation holds, the user confirms that his partial private key is valid.

Then the user picks $x_{ID} \in_R Z_q^*$ randomly and sets x_{ID} as the secret value.

Finally the full private key of the user is (x_{ID}, S_{ID}).

3) **IO_OffSign:**

Given the private key (x_{ID}, S_{ID}), the signer picks $r \in_R Z_q^*$ randomly and computes the offline signature triple (R, V, K).

$$\begin{aligned} R &= rP, \\ V &= S_{ID} + x_{ID}, \\ K &= x_{ID}P + Y. \end{aligned}$$

Please note that these offline components may be computed when the device is idle, because these components are independent of the messages.

4) **IO_OnSign:**

Given a message m and the offline signature triple (R, V, K), the signer computes the online signature

$$\begin{aligned} h &= H_1(m, ID, K, R), \\ \sigma &= r + hV (\mathrm{mod} q). \end{aligned}$$

Finally, the full signature is the triple (K, h, σ).

5) **IO_Verify:**

Given an online/offline signature triple (K, h, σ) on the message m for an identity ID, the verifier computes

$$\begin{aligned} Q_{ID} &= H_0(ID, Y), \\ R' &= \sigma P - h(K + Q_{ID}P_{Pub}), \quad (3) \\ h' &= H_1(m, ID, K, R). \quad (4) \end{aligned}$$

Then he checks whether the following equation holds or not,

$$h' = h. \quad (5)$$

If the equation holds, the signature is valid; otherwise it is invalid.

Correction: We prove that the proposed scheme is correct as follows:

$$\begin{aligned} R' &= \sigma P - h(K + Q_{ID}P_{Pub}) \\ &= (r + hV)P - h(K + Q_{ID}P_{Pub}) \\ &= (r + h(S_{ID} + x_{ID}))P - h(K + Q_{ID}P_{Pub}) \\ &= (r + h(S_{ID} + x_{ID}))P - h(x_{ID}P + Y + Q_{ID}P_{Pub}) \\ &= (r + h(k + sQ_{ID} + x_{ID}))P - h(x_{ID}P + kP + Q_{ID}P_{Pub}) \\ &= (r + h(k + sQ_{ID} + x_{ID}))P - h(x_{ID}P + kP + sQ_{ID}P) \\ &= (r + h(k + sQ_{ID} + x_{ID}))P - h(x_{ID} + k + sQ_{ID})P \\ &= rP \\ &= R. \end{aligned}$$

By $R' = R$, and Equation (4), we can easily get the verification equation (i.e. Equation (5)) is correct.

5 Security Proof

In this section, we show that our proposed scheme is secure.

The security proof of our scheme satisfies existentially unforgeable under adaptive chosen message attacks and ID attacks.

If PKG wants to simulate a user to sign a message, then the types of his behaviors may be as follows:

1) The PKG can forge user's signature without knowing the secret value and the partial private key of the target user.

2) The PKG can forge user's signature that has the knowledge of the master key or the partial private key, but does not have the secret value of the target user.

3) The PKG can forge user's signature that has the knowledge of the secret value and the partial private key of the target user. (But here, the secret value is replaced by PKG, rather than the truly secret value.)

According to the behaviors that describe above, we define the corresponding adversary into three classes. And then shows the following proofs that our scheme is secure.

Theorem 1. *There exists a adversary A_1 who can satisfies existential unforgeability against adaptive chosen ID and message attacks without knowing the user's secret value and the partial private key in random oracle model under the ECDLP assumption.*

Proof. Let A_1 be an adversary who can break our improved scheme. We show how A_1 can be used by a Probabilistic Polynomial Time (PPT) algorithm C to solve the ECDLP. □

Suppose that (P, aP) as a random ECDLP stance of a group G and outputs a. Algorithm C will do the following simulations by interacting with A_1.

Setup: In this stage, Algorithm C performs as follows.

1) Algorithm C sets the public key $P_{Pub} = aP$ and publishes the system parameters $params = \{\lambda, q, P, G, P_{Pub}, H_0, H_1\}$.
2) For $1 \leq u \leq q_{H_0}$, Algorithm C chooses ID_u randomly as the challenge identity for this game, where q_{H_0} denotes the maximum number of querying H_0 oracle.
3) Algorithm C chooses $Q_u \in_R {Z_q}^*$ randomly, sets $Y_u = -Q_u(aP)$, defines $H_0(ID_u, Y_u) = Q_u$, and then adds (ID_u, Y_u, Q_u) on the ${H_0}^{list}$.
4) Algorithm C gives A_1 system parameters $params = \{\lambda, q, P, G, P_{Pub}, H_0, H_1\}$.

Then C starts by answering queries from A_1 as follows.

H_0 **Queries:** A_1 inputs (ID_i, Y_i), and Algorithm C calls the ${H_0}^{list}$ list. If the list ${H_0}^{list}$ contains (ID_i, Y_i, Q_i), C returns to A. Otherwise, C chooses $Q_i \in_R {Z_q}^*$ randomly, adds (ID_i, Y_i, Q_i) to the list ${H_0}^{list}$, and returns Q_i to A.

H_1 **Queries:** A_1 inputs (m_i, ID_i, K_i, R_i), and Algorithm C calls the ${H_1}^{list}$ list. C scans the list ${H_1}^{list}$ to check whether has already been defined. Otherwise, C picks $h_i \in_R {Z_q}^*$ randomly, adds $(m_i, ID_i, K_i, R_i, h_i)$ to the list, and returns h_i to A_1.

Key Extract Queries: We split this query into two phases: the secret extract value and the partial private key extract queries.

Partial private key extract queries: when A_1 requests the private key associated with the identity ID_i, C checks whether the equation of $ID_i = ID_u$ holds or not and maintains the E^{list} list.

1) If $ID_i = ID_u$, then C halts and outputs "failure".
2) If $ID_i \neq ID_u$, C picks $x_{ID_i} \in_R {Z_q}^*$ randomly as the secret value associated with the identity ID_i. Then C chooses $S_{ID_i} \in_R {Z_q}^*$ and computes $K_i = S_{ID_i}P + x_{ID_i}P - Q_i aP$, where $Q_i = H_0(ID_i, Y_i)$. If $H_0(ID_i, Y_i, Q_i)$ has already been defined, then C halts and outputs "failure" (denote the event by E_1). C adds (ID_i, Y_i, Q_i) and $(ID_i, x_{ID_i}, S_{ID_i})$ to the list.. and E^{list}, respectively. Finally, C returns K_i, S_{ID_i}.The probability of the event E_1 is at most $(q_{H_0}+q_E)/2^{\lambda+1}$, where q_E denotes the number of querying key extraction oracle.

Secret value extract queries: If $ID_i = ID_u$, then C halts and outputs "failure"; otherwise finds $(ID_i, x_{ID_i}, S_{ID_i})$ from the list E^{list} and returns associated secret value x_{ID}.

Signing Queries: Assume queries a signature for an identity ID and a message m.

1) If $ID_i = ID_u$, then C chooses $\sigma_u, h_u \in_R {Z_q}^*$ randomly, sets $K_u = aP - Q_u(aP)$ and computes $R_u = \sigma_u P - h_u(K_u + Q_u P_{Pub})$, where $h_u = H_1(m_i, ID_u, K_u, R_u)$. If $H_1(m_i, ID_u, K_u, R_u)$ has already been defined, then C halts and outputs "failure" (denote the event by E_2). Finally, C returns (K_u, h_u, σ_u) as a signature. The probability of the event E_2 is at most $(q_{H_1} + q_S)/2^{\lambda}$, where q_s denotes the maximum number of querying signing oracle.
2) If $ID_i \neq ID_u$, the signature is ordinary, because of C has the secret value and the partial private key. That is, to say C can perform online signing algorithm normally and generate online signature accordingly.

Forgery: Suppose that A_1 outputs a forgeable signature (K^*, h^*, σ^*) on a message m^* for an identity ID^*. Here, ID^* is not submitted to partial private key extract oracle and secret value extract oracle, and (m^*, ID^*) is not query to signing oracle.

1) If $ID^* \neq ID_u^*$ and $K^* \neq K_u^*$, then C halts and outputs "failure" (denote the event by E_3). The probability of the event E_3 is not less than $1/q_{H_0}$.
2) Otherwise, according to forking lemma, there exists an algorithm B which generates two valid signatures $(m^*, ID_u, K_u, R, h_1, \sigma_1)$ and $(m^*, ID_u, K_u, R, h_2, \sigma_2)$ in PPT, where $h_1 \neq h_2$ and Q_u is steadiness due to $H_0(ID_u, Y_u) = Q_u$. So the equations hold as follows:

$$R = \sigma_1 P - h_1(K_u + Q_u P_{Pub}) \text{ by } Equation\ (3)$$

$$R = \sigma_2 P - h_2(K_u + Q_u P_{Pub}) \text{ by } Equation\ (3).$$

After the division, we get $(\sigma_1 - \sigma_2)P = (h_1 - h_2)aP$, then obtain $a = (\sigma_1 - \sigma_2)/(h_1 - h_2)$, and return a as the solution to the ECDLP instance.

Theorem 2. *There exists an adversary who can satisfy existential unforgeability on adaptively chosen ID and message attacks. And has the knowledge of the master key or partial private key, but does not have the user's secret value in random oracle model under the ECDLP assumption.*

Proof. Let A_2 be an adversary who can break our improved scheme. We show how A_2 can be used by a PPT algorithm C to solve the ECDLP. □

Suppose that (P, aP) as a random ECDLP stance of a group G and outputs a. Algorithm C will do the following simulations by interacting with A_2.

Setup: In this stage, Algorithm C performs as follows.

1) Algorithm C sets the public key $P_{Pub} = aP$ and publishes the system parameters, $params = \{\lambda, q, P, G, P_{Pub}, H_0, H_1\}$.
2) For $1 \leq u \leq q_{H_0}$, Algorithm C chooses ID_u randomly as the challenge identity for this game, where q_{H_0} denotes the maximum number of querying H_0 oracle.
3) Algorithm C gives system parameters $params = \{\lambda, q, P, G, P_{Pub}, H_0, H_1\}$ and the master key s.

Then C starts by answering queries from A_2 as follows.

H_0 **Queries:** A_2 inputs (ID_i, Y_i), Algorithm C calls the ${H_0}^{list}$ list. If the list ${H_0}^{list}$ contains (ID_i, Y_i, Q_i), C returns Q_i to A. Otherwise, C chooses $Q_i \in_R {Z_q}^*$ randomly, adds (ID_i, Y_i, Q_i) to the list ${H_0}^{list}$ and returns Q_i to A_2.

H_1 **Queries:** A_2 inputs (m_i, ID_i, K_i, R_i), and Algorithm C calls the ${H_1}^{list}$ list. C scans the list to check whether has already been defined. Otherwise, C picks $h_i \in_R {Z_q}^*$ randomly, adds $(m_i, ID_i, K_i, R_i, h_i)$ to the list ${H_1}^{list}$, and returns h_i to A_2.

Key Extract Queries: We split this query into two phases: the secret value extract and the partial private key extract queries. Since A_2 only knows user's partial private key without knowing the secret key, this phase can leave out the secret value queries.

Partial private key extract queries: when A_2 requests the private key associated with identity ID_i, C checks whether $ID_i = ID_u$ holds or not and maintains the E^{list} list.

1) If $ID_i = ID_u$, then C sets $Y_i = aP$ and finds (ID_i, Y_i, Q_i) from the ${H_0}^{list}$ list. C chooses $k_i \in_R {Z_q}^*$ randomly and computes $S_{ID_i} = k_i + sQ_i$, then adds $(ID_i, S_{ID_i}, \perp)$ to the list (ID_i, S_{ID_i}, k_{i1}) ($\perp$ denotes the unknown secret value for ID_i). Finally, returns S_{ID_i}.
2) If $ID_i \neq ID_u$, C finds (ID_i, Y_i, Q_i) from the ${H_0}^{list}$ list. Then C picks $k_{i1}, k_{i2} \in_R {Z_q}^*$ randomly and computes $S_{ID_i} = k_{i2} + sQ_i$. Add (ID_i, S_{ID_i}, k_{i1}) to the K_i list. Finally, C returns S_{ID_i}.

Signing Queries: Assume A_2 queries a signature for an identity ID and a message m.

1) If $ID_i = ID_u$, then C chooses $\sigma_i, h_i \in_R {Z_q}^*$ randomly, sets $Y_i = aP$, and finds (ID_i, Y_i, Q_i) from the list ${H_0}^{list}$, and further C sets $K_i = Y_i = aP$ and computes $R_i = \sigma_i P - h_i(K_i + Q_i P_{Pub})$, where $h_i = H_1(m_i, ID_i, K_i, R_i)$. If $H_1(m_i, ID_i, K_i, R_i)$ has already been defined, then C halts and outputs "failure" (denote the event by E_2). Finally, C returns (K_i, h_i, σ_i) as an online signature. The probability of the event E_2 is at most $(q_{H_1} + q_S)/2^{\lambda}$, where q_S denotes the maximum number of querying signing oracle.
2) If $ID_i \neq ID_u$, the signature is ordinary, because of C has the secret value and the partial private key. That's to say C can perform online signing algorithm normally and generate online signature accordingly.

Forgery: Suppose that A_2 outputs a forgeable signature (K^*, h^*, σ^*) on a message m^* for an identity ID^*. Here, ID^* is not submitted to secret value extract oracle, and (m^*, ID^*) is not query to signing oracle.

1) If $ID^* \neq ID_u^*$ and $K^* \neq K_u^*$, then C halts and outputs "failure" (denote the event by E_3). The probability of the event E_3 is not less than $1/q_{H_0}$.
2) Otherwise, according to forking lemma, there exists an algorithm B generates two valid signatures $(m^*, ID_u, K_u, R, h_1, \sigma_1)$ and $(m^*, ID_u, K_u, R, h_2, \sigma_2)$ in PPT, where $h_1 \neq h_2$ and $Y' = K'P$ is steadiness. So the equations hold as follows:

$$R = \sigma_1 P - h_1(K_u + Q_u P_{Pub}) \text{ by } Equation\ (3)$$

$$R = \sigma_2 P - h_2(K_u + Q_u P_{Pub}) \text{ by } Equation\ (3).$$

After the division, we get $(\sigma_1 - \sigma_2)P = (h_1 - h_2)(a + sQ_u)P$, then obtain $a = (\sigma_1 - \sigma_2)/(h_1 - h_2) - sQ_u$, and return a as the solution to the ECDLP instance.

Theorem 3. *If PKG simulates a legitimate user to forge the signature, who has the knowledge of the user's secret value and the partial private key (the secret value is not real, which represents an alternative), then we can prove to the intercessor that PKG mentioned above is dishonest.*

Proof. According to the Theorem 1 and Theorem 2, our scheme is unforgeable for the honest PKG or the negative dishonest PKG. We split this process into two steps: to forge private key and sign message. □

1) Forge Private Key.

Assume that ID as user's identity and (x_{ID}, S_{ID}) as the private key. Then PKG simulates the user to sign messages by two ways:

a. Knowing the user's secret value x_{ID};
b. Replacing the user's secret value x_{ID}. However, the user chooses x_{ID} randomly and thus it is impossible for PKG to get x_{ID}.

Thus, PKG only chooses to replace the user's secret value x_{ID} and generates another private key. The details steps as follows:

Table 1: Notions and definitions of time complexities

Notions	Definitions
T_M	Time required for executing a modular multiplication operation.
T_{ZM}	Time required for executing a multiplication operation in Z_q^*.
T_{GM}	Time required for executing a multiplication operation in group.
T_{GE}	Time required for executing an exponentiation operation in group.
T_H	Time required for executing a hash operation.
T_E	Time required for executing a modular exponentiation operation, $T_E \approx 240T_M$.
T_P	Time required for executing a bilinear pairing operation,$T_P \approx 87T_M$.
T_{PE}	Time required for executing an pairing-based exponentiation operation, $T_{PE} \approx 43.5T_M$.
T_{PM}	Time required for executing an elliptic curve scalar point multiplication operation, $T_{PM} \approx 29T_M$.
T_{MTP}	Time required for executing a hash function operation, $T_{MTP} \approx T_{PM} \approx 29T_M$.
T_{PA}	Time required for executing a point addition operation of two elliptic curve points, $T_{PA} \approx 0.12T_M$.

a. PKG chooses x_{ID} to replace the user's secret value (the probability of $x_{ID'} = x_{ID}$ can $S_{ID'} = k' + sQ_{ID'}(\mathrm{mod}q)$ be ignored).

b. PKG picks $k' \in_R {Z_q}^*$ randomly and computes $Y' = k'P$ and $S'_{ID} = k' + sQ'_{ID}\,(\mathrm{mod}q)$, where $Q_{ID'} = H_0(ID, Y')$. Assume that Y', S'_{ID} satisfy the Equation (2), then it outputs the private key (x'_{ID}, S'_{ID}).

2) Sign Message.

After PKG forging the user's private key $(x_{ID'}, S_{ID'})$, he starts to perform signing algorithm. (k', h', σ') denotes the signature on a message m for the user. The user can execute signing algorithm twice to prove whether (k', h', σ') is forged by PKG or an adversary colluded with PKG. Suppose that the user generates two signatures (K, h_1, σ_1) and (K, h_2, σ_2), and submits (K, h_1, σ_1) and (K, h_2, σ_2) to the intercessor TA.

But here $K' \neq K$. If PKG wants to make $K' = K$, then PKG needs to hold the equation $(k' + x'_{ID})P = (k + x_{ID})P$. Further PKG requires to know the value $Y' = (k + x_{ID} - x'_{ID})P = k'P$, but PKG does not know x_{ID}. According to the ECDLP, PKG can't obtain k, Q_{ID} and S_{ID}, so $K' \neq K$.

If three signatures above are valid, then K in (K, h_1, σ_1) and (K, h_2, σ_2) is same. We get $K' \neq K$ in (K', h', σ'), so (K', h', σ') must be forged by PKG or an adversary colluded with PKG.

In summary, the Theorem 1 and Theorem 2 can make our scheme unforgeable for the honest PKG or the negative dishonest PKG. Only PKG knows the master key s, and thus only PKG can replace the user's secret value to generate another private key (x'_{ID}, S'_{ID}). According to the above, PKG can not make $K' = K$, so we can prove whether PKG is honest or not by checking the equation $K' = K$ or not.

6 Performance Evaluation

In this section, we evaluated the performance of our scheme and gave a detailed comparison with other schemes proposed in the literatures [9, 13, 18, 19, 26, 27, 28] proposed in the literature. To estimate the operating overhead, we define the notations in Table 1. Please note that the time of other light operations is ignored in the comparisons (e.g., addition operation in Z_q^*), since it is relatively smaller.

In most online/offline signatures, the main computational costs are shifted to the offline phase, so the efficiency is dependent on the online and verification phase. The comparisons of our scheme with other online/offline schemes [9, 13, 18, 19, 26, 27, 28] are listed in Table 2. It is obvious that our scheme does not require any bilinear pairing and MTP hush function operations. Therefore, it is more efficient than these schemes [9, 13, 18, 24, 28] in terms of computational costs in the online signing, and also more efficient than other schemes [9, 13, 18, 19, 26, 27, 28] in terms of computational costs in the verification.

In Table 3[1], the comparisons for different aspects between our scheme and other related schemes are summarized. This table shows that our scheme supports all these good attributes, but the schemes proposed in [9, 13, 18, 19, 26, 27, 28] do not.

To sum up, our identity-based online/offline signature scheme enjoys the following good advantages: (1) No bilinear pairing and probabilistic MTP function, (2) Low computation costs, (3) No key escrow problem, (4) Key confirmation, (5) Provable security under the random oracle model against adaptive chosen ID and message attacks, and (6) supports traceability.

[1]Note: A_1, -based cryptosystem; A_2, Hardness problems; A_3, Random oracle; A_4, No bilinear pairing and MTP hash function; A_5, Low computational costs; A_6, No key escrow problem; A_7, Key confirmation; A_8, Provable security; A_9, Traceability; IBC, Identity-Based Cryptosystem; CLC, Certificateless Cryptosystem; CDHP, Computation Diffie-Hellman Problem; GDH, Gap Diffie-Hellman Problem; IFP, Integer Factorization Problem; ECDLP, Elliptic Curve Discrete Logarithm Problem.

Table 2: Computational cost comparison of the proposed scheme with others

Schemes	**The online signing phase**	**The verification phase**	**The total computational cost**
Xu et al. [27]	$1T_{ZM}+1T_H$	$2T_{PM}+2T_P+1T_{MTP}$ $+1T_{PA}+1T_H$ $\approx 261.12T_M+1T_H$	$\approx 261.12T_M+1T_{ZM}+2T_H$
Xu et al. [26]	$1T_{ZM}+1T_H$	$2T_{PM}+2T_P+1T_{MTP}$ $+2T_{PA}+1T_H$ $\approx 261.24T_M+1T_H$	$\approx 261.24T_M+1T_{ZM}+2T_H$
Ge et al. [9]	$2T_E+2T_M+1T_H$ $\approx 482T_M+1T_H$	$4T_E+2T_M+4T_{GM}+3T_H$ $\approx 962T_M+4T_{GM}+3T_H$	$\approx 1444T_M+4T_{GM}+4T_H$
Wu et al. [25]	$1T_{ZM}+1T_H$	$2T_{PM}+2T_P+1T_{MTP}$ $+2T_{PA}+1T_H$ $\approx 261.24T_M+1T_H$	$\approx 261.24T_M+1T_{ZM}+2T_H$
Ming et al. [28]	$2T_{ZM}+1T_H$	$3T_{PM}+2T_P+1T_{MTP}$ $+1T_{PA}+2T_H$ $\approx 290.12T_M+2T_H$	$\approx 290.12T_M+2T_{ZM}+3T_H$
Selvi et al. [18]	$2T_{ZM}+2T_H$	$3T_{GM}+4T_{GE}+4T_H$	$\approx 3T_{GM}+3T_{GE}$ $+2T_{ZM}+6T_H$
Selvi et al. [19]	$1T_{ZM}+1T_H$	$6T_{PM}+4T_{PA}+4T_P$ $+3T_{MTP}+2T_{ZM}+3T_H$ $\approx 609.48T_M+2T_{ZM}$	$\approx 609.48T_M+3T_{ZM}+4T_H$
Wang et al. [24]	$(n-1)T_{ZM}$	$3T_E+2T_{ZM}$ $\approx 720T_M+2T_{ZM}$	$\approx 720T_M+(n+1)T_{ZM}$
Kar et al. [13]	$2T_{PM}+2nT_{ZM}+1T_H$ $\approx 58T_M+2nT_{ZM}+1T_H$	$3T_P+1T_{PE}+1T_{GM}$ $+1T_{MTP}+1T_H$ $\approx 304.5T_M+1T_{GM}$	$\approx 362.62T_M+1T_{GM}$ $+2nT_{ZM}+2T_H$
Beak et al. [4]	$(n-1)T_{ZM}+1T_{GE}$ $+1T_H+1T_M$	$3T_{GE}+3T_{GM}+1T_H$	$(n-1)T_{ZM}+4T_{GE}+3T_{GM}$ $+2T_H$
Abinav et al. [1]	$1T_{ZM}+1T_H$	$6T_{PM}+4T_{PA}+3T_H$ $+1T_{MTP}$	$\approx 203.48T_M+4T_H+1T_{ZM}$
Aboud et al. [2]	$(n-1)T_{ZM}+1T_H+1T_M$	$3T_P+1T_{PE}+1T_{GM}$ $+1T_{MTP}+1T_H$ $\approx 304.5T_M+1T_{GM}$	$\approx 3T_{GE}+(n+1)T_{ZM}$ $+2T_H+1T_M$
Our Scheme	$1T_{ZM}+1T_H$	$3T_{GE}+2T_{ZM}+1T_H+1T_M$	$\approx 87.24T_M+1T_{ZM}+3T_H$

Table 3: Comparisons of the proposed scheme with the existing schemes for different attributes

Schemes	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9
Xu et al. [27]	IBC	CDHP	Yes	No	No	No	No	No	No
Xu et al. [26]	IBC	CDHP	Yes	No	No	No	No	No	No
Ge et al. [9]	CLC	ECDLP	Yes	Yes	No	Yes	Yes	Yes	No
Wu et al. [25]	IBC	CDHP	Yes	No	No	No	No	Yes	No
Ming et al. [28]	IBC	CDHP	Yes	No	No	No	No	Yes	No
Selvi et al. [18]	CLC	GDHP	No	Yes	No	Yes	Yes	No	No
Selvi et al. [19]	CLC	GDHP	Yes	No	No	Yes	Yes	Yes	No
Wang et al. [24]	IBC	IFP	No	Yes	No	No	No	No	No
Kar et al. [13]	IBC	CDHP	Yes	No	No	No	No	Yes	No
Beak et al. [4]	IBC	ECDLP	Yes	Yes	No	No	Yes	Yes	No
Abinav et al. [1]	CLC	CDH	Yes	No	No	Yes	Yes	Yes	No
Aboud et al. [2]	IBC	ECDLP	Yes	Yes	No	Yes	Yes	Yes	No
Our Scheme	IBC	ECDLP	Yes	Yes	Yes	Yes	Yes	Yes	Yes

7 Conclusion

ID-based online/offline signature scheme is a combination of IBC and online/offline signature. It has the following advantages: (1) eliminating the costly certificate verification process and the storage of the length certificate, (2) producing some computation results in the offline phase which is stored in advance and later used when the message to be signed is known, such that a valid signature can be generated quickly in the online phase. In this paper, we designed an bilinear paring and MTP function-free identity-based online/offline signature scheme without employing complex bilinear paring and MTP function, which was proven to satisfy the existential unforgeablility against adaptively chosen message and ID attacks in the random oracle under the ECDLP assumption. Our proposed scheme is computationally more efficient than other related signature schemes. Especially, our scheme provides Key confirmation and Traceability. Next, applying the proposed scheme in some resource-constrained environments is our future work.

Acknowledgments

This work was supported by National Natural Science Foundation of China (Nos 61173187, 11301002 and 61572001), Natural Science Foundation of Anhui Province (No 1408085QF107), the 211 Project of Anhui University (Nos 33190187 and 17110099), and the Educational Commission of Anhui Province, China (KJ2015A326).

References

[1] K. Abinav, S. Badrinarayanan, C. P. Rangan, S. S. D. Selvi, S. S. Vivek, and V. K. Pradhan, "A revocable online-offline certificateless signature scheme without pairing," *IACR Cryptology ePrint Archive*, vol. 2013, pp. 758, 2013.

[2] S. J. Aboud, "Secure online-offline identity-typed signature scheme," *International Journal of Scientific Research and Management Studies*, vol. 1, pp. 2349–3371, 2015.

[3] S. S. Al-Riyami and K. G. Paterson, "Certificateless public key cryptography," in *Advances in cryptology (ASIACRYPT'03)*, pp. 452–473, 2003.

[4] J. Baek, Y. J. Byon, E. Hableel, and M. Al-Qutayri, "An authentication framework for automatic dependent surveillance-broadcast based on online/offline identity-based signature," in *2013 Eighth International Conference on P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC'13)*, pp. 358–363, 2013.

[5] D. Boneh and M. Franklin, "Identity-based encryption from the weil pairing," in *Advances in Cryptology (CRYPTO'01)*, pp. 213–229, 2001.

[6] X. Chen, F. Zhang, W. Susilo, and Y. Mu, "Efficient generic on-line/off-line signatures without key exposure," in *Applied Cryptography and Network Security*, pp. 18–30, 2007.

[7] Y. Desmedt and J. J. Quisquater, "Public-key systems based on the difficulty of tampering (is there a difference between DES and RSA)," in *Advances in Cryptology (CRYPTO'86)*, pp. 111–117, 1986.

[8] S. Even, O. Goldreich, and S. Micali, "Online/offline digital signatures," in *Advances in Cryptology (CRYPTO'90)*, pp. 263–375, 1990.

[9] A. Ge, S. Chen, and X. Huang, "A concrete certificateless signature scheme without pairings," in *MInternational Conference on ultimedia Information Networking and Security (MINES'09)*, vol. 2, pp. 374–377, 2009.

[10] M. C. Gorantla and A. Saxena, "An efficient certificateless signature scheme,". in *Computational Intelligence and Security*, pp. 110–116. 2005.

[11] S. K. H. Islam and G. P. Biswas, "A pairing-free identity-based authenticated group key agreement protocol for imbalanced mobile networks," *Annals of Telecommunications*, vol. 67, no. 11-12, pp. 547–558, 2012.

[12] S. K. H. Islam and G. P. Biswas, "Provably secure and pairing-free certificateless digital signature scheme using elliptic curve cryptography," *International Journal of Computer Mathematics*, vol. 90, no. 11, pp. 2244–2258, 2013.

[13] 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.

[14] A. V. N. Krishna, A. H. Narayana, K. M. Vani, "Window method based cubic spline curve public key cryptography," *International Journal of Electronics and Information Engineering*, vol. 4, no. 2, pp. 94–102, 2016.

[15] X. X. Li, K. F. Chen, and L. Sun, "Certificateless signature and proxy signature schemes from bilinear pairings," *Lithuanian Mathematical Journal*, vol. 45, no. 1, pp. 76–83, 2005.

[16] U. M. Maurer and Y. Yacobi, "Non-interactive public-key cryptography," in *Advances in Cryptology (EUROCRYPT'91)*, pp. 498–507, Springer, 1991.

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

[18] S. S. D. Selvi, S. S. Vivek, V. K. Pradhan, and C. P. Rangan, "Efficient certificateless online/offline signature," *Journal of Internet Services and Information Security*, vol. 2, no. 3/4, pp. 77–92, 2012.

[19] S. S. D. Selvi, S. S. Vivek, V. K. Pradhan, and C. P. Rangan, "Efficient certificateless online/offline signature with tight security," *Journal of Internet Services and Information Security*, vol. 1, no. 1/2, pp. 115–137, 2013.

[20] A. Shamir, "Identity-based cryptosystems and signature schemes," in *Advances in Cryptology*, pp. 47–53, Springer, 1984.

[21] A. Shamir and Y. Tauman, "Improved online/offline signature schemes," in *Advances in Cryptology (CRYPTO'01)*, pp. 355–367, Springer, 2001.
[22] H. Tanaka, "A realization scheme for the identity-based cryptosystem," in *Advances in Cryptology (CRYPTO'87)*, pp. 340–349, Springer, 2006.
[23] S. Tsujii and T. Itoh, "An id-based cryptosystem based on the discrete logarithm problem," *IEEE Journal on Selected Areas in Communications*, vol. 7, no. 4, pp. 467–473, 1989.
[24] Z. Wang and W. Chen, "An id-based online/offline signature scheme without random oracles for wireless sensor networks," *Personal and Ubiquitous Computing*, vol. 17, no. 5, pp. 837–841, 2013.
[25] T. S. Wu, Y. S. Chen, and K. Y. Lin, "Id-based online/offline signature from pairings," in *International Computer Symposium (ICS'10)*, pp. 198–203, 2010.
[26] S. Xu, Y. Mu, and W. Susilo, "Online/offline signatures and multisignatures for aodv and dsr routing security," in *Information Security and Privacy*, pp. 99–110, 2006.
[27] S. Xu, Y. Mu, and S. Willy, "Efficient authentication scheme for routing in mobile ad hoc networks," in *Embedded and Ubiquitous Computing (EUC'05)*, pp. 854–863, 2005.
[28] M. Yang and Y. Wang, "Improved identity based online/offline signature scheme," in *Ubiquitous Intelligence & Computing and 7th International Conference on Autonomic & Trusted Computing (UIC/ATC'10)*, pp. 126–131, 2010.
[29] W. S. Yap, S. H. Heng, and B. M. Goi, "An efficient certificateless signature scheme," in *Emerging Directions in Embedded and Ubiquitous Computing*, pp. 322–331, 2006.
[30] Z. Zhang, D. S. Wong, J. Xu, and D. Feng, "Certificateless public-key signature: security model and efficient construction," in *Applied Cryptography and Network Security*, pp. 293–308, 2006.

Dan Liu is a Master Candidate at the School of Computer Science and Technology, Anhui University, China. Her research interests include digital signature, security and privacy for mobile wireless networks privacy protection, etc..

Shun Zhang is currently an Associate Professor and Master Advisor at the School of Computer Science and Technology, Anhui University, China. His research interests include secure multi-party computation, big data, etc..

Hong Zhong is a Professor and PhD Advisor at the School of Computer Science and Technology, Anhui University, China. Her research interests include security protocols, wireless sensor networks, etc..

Runhua Shi received the PhD degree from University of Science and Technology of China in 2011. He is currently a Professor with Anhui University, and a visiting fellow at the School of Computer Science and Software Engineering, University of Wollongong. His current research interest includes classical and quantum cryptography, in particular, privacy-preserving multiparty computations.

Yimin Wang is a PhD Candidate at the School of Computer Science and Technology, Anhui University, China. His research interests include security and privacy for wireless networks, cloud computing, big data, etc..