3-1-2017

Efficient Revocable ID-Based Signature With Cloud Revocation Server

Xiaoying Jia
South-Central University for Nationalities, China

Debiao He
Wuhan University, China

Sherali Zeadally
University of Kentucky, szeadally@uky.edu

Li Li
Wuhan University, China

Efficient Revocable ID-Based Signature With Cloud Revocation Server

Notes/Citation Information

Published in *IEEE Access*, v. 5, p. 2945 - 2954.

Digital Object Identifier (DOI)
https://doi.org/10.1109/ACCESS.2017.2676021

SPECIAL SECTION ON FUTURE NETWORKS: ARCHITECTURES, PROTOCOLS, AND APPLICATIONS

IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received February 1, 2017, accepted February 21, 2017, date of publication March 1, 2017, date of current version March 28, 2017.

Digital Object Identifier 10.1109/ACCESS.2017.2676021

Efficient Revocable ID-Based Signature With Cloud Revocation Server

XIAOYING JIA[1], DEBIAO HE[2,3], SHERALI ZEADALLY[4], AND LI LI[5]

[1]School of Mathematics and Statistics, South-Central University for Nationalities, Wuhan 430074, China
[2]State Key Lab of Software Engineering, School of Computer, Wuhan University, Wuhan 430072, China
[3]Guangxi Key Laboratory of Cryptography and Information Security, Guilin University of Electronic Technology, Gulin 541000, China
[4]College of Communication and Information, University of Kentucky, Lexington, KY 40506, USA
[5]International School of Software, Wuhan University, Wuhan 430072, China

Corresponding author: L. Li (lli@whu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61402339, Grant 61572379, Grant 61501333, and Grant 61603419 and in part by the Fundamental Research Funds for the Central Universities under Grant CZY15018.

ABSTRACT Over the last few years, identity-based cryptosystem (IBC) has attracted widespread attention because it avoids the high overheads associated with public key certificate management. However, an unsolved but critical issue about IBC is how to revoke a misbehaving user. There are some revocable identity-based encryption schemes that have been proposed recently, but little work on the revocation problem of identity-based signature has been undertaken so far. One approach for revocation in identity-based settings is to update users' private keys periodically, which is usually done by the key generation center (KGC). But with this approach, the load on the KGC will increase quickly when the number of users increases. In this paper, we propose an efficient revocable identity-based signature (RIBS) scheme in which the revocation functionality is outsourced to a cloud revocation server (CRS). In our proposed approach, most of the computations needed during key-updates are offloaded to the CRS. We describe the new framework and the security model for the RIBS scheme with CRS and we prove that the proposed scheme is existentially unforgeable against adaptively chosen messages and identity attacks in the random oracle model. Furthermore, we monstrate that our scheme outperforms previous IBS schemes in terms of lower computation and communication costs.

INDEX TERMS Identity-based signature, revocation, cloud computing, outsourcing.

I. INTRODUCTION

Digital signature is a critical feature of public key cryptography that provides user identification, authentication and non-repudiation. In traditional public key infrastructure (PKI), users' public keys used to verify signatures are bound with their certificates. Certificate authorities (CAs) are responsible for issuing, maintaining and revoking certificates. In identity-based cryptosystems, however, a user's identity information is the public key. It is a challenge to verify if a user has been revoked or not. In 2001, Boneh and Franklin [1] suggested that the key generator center (KGC) updates secret keys for all non-revoked users periodically. The idea was adopted by many identity-based encryption schemes to realize revocation functionality. Unfortunately, there are two drawbacks with their proposals. First, the KGC must kept online, which brings out some security threats. Second, the overhead at the KGC will dramatically increase as the number of users increases.

With the rapid development of cloud computing, many organizations tend to outsource computation tasks to some powerful cloud based server. In fact, it is not rare in the history of cryptography to outsource heavy computation tasks to a third party. Quite recently Li *et al.* [2] introduced an approach to outsource the key-updating tasks to a Key Update Cloud Server Provider (KU-CSP) and proposed an efficient revocable identity-based encryption (IBE) scheme. We adopt their approach and apply it to identity-based signature (IBS) settings. A trivial idea is to offload all key-updating tasks to the cloud server. However, there are some security issues we must take into account: the cloud server is not always trusted. So we split the signing key of a user into an initial identity key and a time update key. The former is a long-term key bound to the user's identity and issued by the KGC, and the latter is a short-term key related not only to the user's identity, but also to the current time period. The time update key is issued and updated by a cloud revocation server (CRS) periodically. The CRS cannot forge a signature because it does not hold the complete signing key. To revoke users, the KGC simply notifies the cloud server to stop issuing new time update keys for them. With this technique, many existing IBS schemes can be improved to be revocable.

A. RELATED WORK

We discuss a few recently proposed IBS schemes below.

Shamir first introduced the idea of identity-based cryptosystem (IBC) [3], after that Fiat and Shamir [4] presented a construction of IBS scheme based on the factoring problem. Since then, several other proposals based on factoring have been developed [5]–[7].

The first fully practical implementation of identity-based setting emerged in 2001, when Boneh and Franklin [1] proposed an IBE scheme using Weil pairing on elliptic curves. Since then many solutions for IBS schemes with bilinear pairings have been proposed. Sakai *et al.* [8] presented an IBS scheme based on bilinear pairings but no security analysis was given. Choon and Cheon [9] proposed an IBS scheme by utilizing gap bilinear Diffie-Hellman (GDH) groups. Paterson [10] presented another efficient IBS scheme and reduced the security of their scheme to a non-IBS scheme. Hess [11], [12] developed an efficient IBS scheme and extended it to a generic framework, from which several variations can be exported (include ElGamal variations and Schnorr version).The author also considered the key escrow by extending the system to multiple trust authorities. Galindo and Garcia [13] also proposed a Schnorr-like lightweight IBS scheme without pairings. Bellare *et al.* [14] provided a framework for security proof of IBS schemes. Zhang *et al.* [15] proposed an efficient IBS scheme secure under the k-CAA assumption. Barreto *et al.* [16] presented an identity-based signcryption (IBSC) scheme with bilinear pairings. Based on Water's IBE scheme [17], Paterson and Schuldt [18] presented an efficient IBS scheme and proved its security in the standard model. In recent years, many IBS schemes such as those based on ring signatures, blind signatures, proxy signatures, group signatures etc. were proposed [19]–[22].

There are several efficient proposals of IBS schemes with or without bilinear pairings which have been proposed. However, only a few of them discussed the revocation of misbehaving users. Boneh and Franklin [1] developed a general approach to implement the revocation functionality in identity-based cryptosystems. That is, the KGC generates new secret keys for each non-revoked user periodically, and it simply stops to issue new private keys for the revoked users. Based on this idea, various revocable IBE schemes were proposed [23]–[27] in the past.

The first revocable IBS scheme was proposed by Tsai *et al.* [28], which adopted the revocation technique employed in [27]. The authors proved the security of their scheme in the standard model. Based on their scheme, Hung *et al.* [29] proposed another RIBS scheme with improved security. Sun *et al.* [30] presented an efficient RIBS scheme without pairing but no security proof was given. Recently, Wei *et al.* [31] proposed a forward secure RIBS scheme employing the complete subtree (CS) method where the KGC must maintain a binary tree on which each node represents a user.

In the above RIBS schemes [28]–[31], the KGC is responsible not only for issuing the initial identity key for each registered user, but also for renewing the time update keys for non-revoked users periodically, which brings two drawbacks. First, the KGC needs to be kept online which is not secure. Second, with the increasing in the number of system users, the computation and communication overheads at the KGC also increase quickly. In this case, the KGC will become the security and performance bottleneck of the whole cyrptosystem.

B. OUR CONTRIBUTIONS

In this work, we propose the first RIBS scheme with a cloud revocation server. We describe the framework of a RIBS scheme with outsourced revocation and formalize the security model. Then we describe our proposed scheme in detail and analyze its security. We prove that our scheme is existentially unforgeable against adaptive chosen message and identity attacks in the random oracle model. Neither a revoked user nor a curious cloud server can forge a valid signature even if they collude with other non-revoked users in the system. We also provide a performance evaluation of our scheme and we compare its performance with other IBS schemes.

C. ORGANIZATION

The rest of the paper is organized as follows. We present preliminary works in Section II. Section III describes the framework of a RIBS scheme with outsourced cloud revocation server and formalizes its security model. Section IV describes our proposed RIBS scheme in detail. We present the security analysis of our scheme in Section V. Section VI presents the performance evaluation results of our scheme. Section VII concludes the paper.

II. PRELIMINARY

A. IDENTITY-BASED SIGNATURE

A typical IBS scheme involves three parties: the KGC, the signer and the verifier. There are four algorithms in an IBS scheme defined as follows.

- **Setup:** $(MSK, PP) \leftarrow Setup(\lambda)$**.** KGC takes as input the security parameter λ, and outputs the master system key MSK and system public parameters PP including the system public key P_{pub} .
- **Initial Key Extraction:** $S_{ID} \leftarrow KeyExt(MSK, ID)$**.** KGC generates a private key for each user. It takes as input MSK and a user's identity ID, and returns the private key S_{ID}.
- **Signing:** $\sigma \leftarrow Sign(m, S_{ID}, PP)$**.** The signer takes as input his/her private key S_{ID}, the message m and the public parameters, and outputs a signature σ.
- **Verification:** $Accept/Reject \leftarrow Ver(\sigma, ID, m, PP)$**.** The verifier takes as input the signature σ, the identity ID of the signer, the message m and PP , and returns an "Accept" or a "Reject" to demonstrate if the signature is valid or not.

The consistency of an IBS scheme requires that for any S_{ID} generated by algorithm *KeyExt* when given ID as input, and for any $\sigma = Sign(m, S_{ID}, PP)$, $Ver(\sigma, ID, m, PP) =$ "*Accept*" holds.

B. BILINEAR PAIRINGS AND COMPUTATIONAL ASSUMPTIONS

Let G be an additive cyclic group, whose order is a large prime q. P is a generator of G. G_T is a multiplicative cyclic group of the same order q. The map $\hat{e} : G \times G \rightarrow G_T$ is said to be an admissible bilinear map if it satisfies:

- Bilinearity: For all $P, Q \in G$, $x, y \in Z_q^*$, there is $\hat{e}(xP, yQ) = \hat{e}(P, Q)^{xy}$;
- Non-degeneracy: There exists $P, Q \in G$ such that $\hat{e}(P, Q) \neq 1_{G_T}$;
- Computability: For any element $P, Q \in G$, there is an polynomial time algorithm to compute $\hat{e}(P, Q) \in G_T$.

Next we present the mathematical assumption used in our scheme.

Computational Diffie-Hellam Problem (CDH): Given a triple (P, aP, bP) for some unknown $a, b \in Z_q^*$, we compute abP.

The CDH assumption says that there is no polynomial time algorithm which can solve the CDH problem with non-negligible probability.

III. SYSTEM FRAMEWORK AND SECURITY MODEL

In this section, we describe the system framework of an outsourced RIBS scheme and its security model.

A. SYSTEM FRAMEWORK

A RIBS scheme involves three parties: the KGC, the CRS and users (signers and verifiers). At the beginning of the system initialization, the KGC generates and publishes some common parameters and sends a secret master time key to the CRS. Then the KGC issues the initial identity key for each user with its master system key when the user is registered. The CRS issues and updates the users' time update keys according to the revocation user list received from the KGC. If a user is in the revocation user list, then the CRS refuses to update the time update key for the user. We present the framework of our system in Fig. 1. Table 1 shows the notations used in the proposed RIBS scheme.

There are five algorithms in a RIBS scheme: system initialization algorithm (*Setup*), initial key extraction algorithm (*InitKeyExt*), time key updating algorithm (*TimeKeyUpd*), signing algorithm (*Sign*), verification algorithm (*Ver*). The KGC maintains a revocation list (RL) which contains the identities of revoked users and the RL is updated periodically.

- $Setup(1^\lambda)$: The KGC takes a security parameter λ and outputs a master system key msk, a master time key mtk, a time period list $T = (T_0, T_1, \ldots)$ and system public parameters PP. KGC keeps msk for itself, and sends mtk to the CRS securely. PP is published to all users in the system.

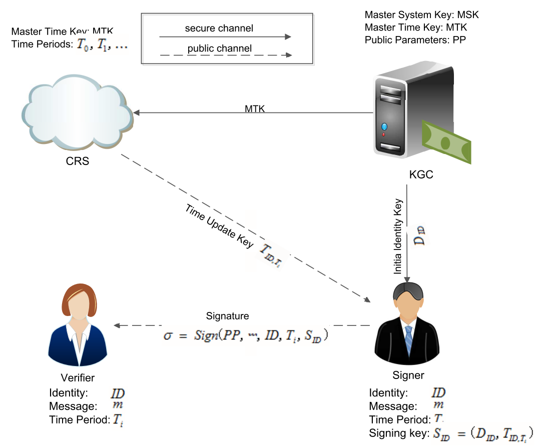

FIGURE 1. **Identity-based signature with (cloud) outsourced revocation.**

TABLE 1. **Summary of notations.**

Notation	Description
ID	identity of a user $ID \in \{0,1\}^*$
T_i	ith time period
PP	system public parameters
msk	master system key
mtk	master time key
S_{ID,T_i}	signing key with identity ID and time period T_i
D_{ID}	initial identity key with identity ID
T_{ID,T_i}	time update key with identity ID and time period T_i
H_i	a map-to-point hash function
h_i	an ordinary hash function

- $InitKeyExt(PP, msk, ID)$: On receiving a register request from the user with identity ID, the KGC runs this algorithm to issue a secret identity key D_{ID} with the input PP, msk, a user's identity ID, and sends D_{ID} to the user securely.
- $TimeKeyUpd(PP, mtk, ID, T_i)$: On receiving an update request from a user, the CRS first checks if the user is in the RL. If it is, the CRS rejects the update request. If not, the CRS runs the algorithm and generates a new time key TK_{ID,T_i} for the user with the input PP, mtk, the user's identity ID and current time period T_i.
  A user's signing key S_{ID,T_i} consists of two parts: $S_{ID,T_i} = (D_{ID}, TK_{ID,T_i})$.
- $Sign(PP, m, ID, T_i, S_{ID,T_i})$: To sign a message m, the signer runs this algorithm with input PP, m, his own identity ID, current time period T_i and S_{ID,T_i}, and outputs the signature σ.
- $Ver(PP, m, \sigma, ID, T_i)$: To verify a signature σ on message m with the signer's identity ID and time period T_i, the verifier runs the algorithm with PP, m, σ, ID and T_i, and outputs an "Accept" or "Reject" according to the validity of the signature.

The consistency criterion states that for any message m, any identity ID and any time period T_i, if $\sigma = Sign(PP, m, ID, T_i, S_{ID,T_i})$, where

$S_{ID,T_i} = (InitKeyExt(msk, ID), TimeKeyUpd(mtk, ID, T_i))$, then we have $Ver(PP, m, \sigma, ID, T_i) =$ "$Accept$" with a high probability.

B. SECURITY MODEL

Bellare *et al.* [14] first formalized the security of an IBS scheme in 2004, namely, security against existential forgery on adaptively chosen message and identity attacks (EUF-CMA). Based on this formalization, we first consider the following two types of adversaries.

- *Type I adversary* A_I. A_I is a revoked user. Suppose A_I has identity ID and was revoked at time period T_i. A_I intends to produce valid signatures after time period T_i. A_I still owns the initial identity key D_{ID}, and we assume that A_I can collude with other legal users to obtain their identity keys and time update keys at arbitrary time periods. A_I cannot know is its own time update keys after time period T_i.
- *Type II adversary* A_{II}. Type II adversary can be seen as a curious CRS who tries to create a valid signature in the name of a system user. Since the CRS holds the master time key, so it can obtain the time update key of any user at any time. We also assume that A_{II} can collude with other users to obtain their identity keys. In this case, the adversary cannot know the target user's identity key D_{ID}.

We define the security model of an outsourced RIBS scheme through the following two games between a challenger and one of the above two types of adversaries. The game between A_I and a challenger S_I is defined below.

Game 1:

- **Setup.** S_I runs the *Setup* algorithm with input security parameter λ, and outputs msk, mtk and PP as defined in the system framework. S_I keeps msk, mtk secret and sends PP to A_I.
- **Query.** A_I makes a series of queries to S_I adaptively, and S_I responds to each type of queries in the following way.
  - *Initial key extract query (ID).* A_I issues this query to get the initial key of some user with identity ID. S_I runs *InitKeyExt* algorithm with input (PP, msk, ID), and returns the resulting D_{ID} to A_I.
  - *Time key update query* (ID, T_i). A_I issues this query to get the time update key of some user with identity ID on time period T_i. S_I runs the *TimeKeyUpd* algorithm with input (PP, mtk, ID, T_i), and returns the resulting TK_{ID,T_i} to A_I.
  - *Signing query* (m, ID, T_i). When A_I issues a signing query with a message m, the identity ID and time period T_i, S_I runs *Sign* algorithm and outputs a signature σ to A_I.
- **Forgery.** At last A_I outputs a tuple $(m^*, ID^*, T_i^*, \sigma^*)$ with the following two constraints:
  1) A_I does not issue any *time key update query* on (ID^*, T_i^*).
  2) σ^* is not returned by a *signing query* on input (m^*, ID^*, T_i^*) issued by A_I.

  It is said that A_I succeeds in attacking the scheme if $Ver(PP, m^*, \sigma^*, ID^*, T_i^*) =$ "$Accept$". A_I 's advantage $Adv_{A_I}(\lambda)$ is defined as

$$Adv_{A_I}(\lambda) = Pr[Ver(PP, m^*, ID^*, T_i^*) = \text{"}Accept\text{"}].$$

The game between adversary A_{II} and a challenger S_{II} is defined as follows.

Game 2: The **Setup** and **Query** phases are the same as in *Game I*.

Forgery. At the end of the **Query** phase, A_{II} outputs a tuple $(m^*, ID^*, T_i^*, \sigma^*)$ with the following two constraints:

1) A_{II} does not issue any *initial key extract query* on input ID^*.
2) σ^* is not returned by a *signing query* on input (m^*, ID^*, T_i^*) issued by A_{II}.

It is said that A_{II} succeeds in attacking the scheme if $Ver(PP, m^*, \sigma^*, ID^*, T_i^*) =$ "$Accept$". A_{II} 's advantage $Adv_{A_{II}}(\lambda)$ is defined as

$$Adv_{A_{II}}(\lambda) = Pr[Ver(PP, m^*, \sigma^*, ID^*, T_i^*) = \text{"}Accept\text{"}].$$

From the above games we have the following security definition of a RIBS scheme.

Definition 1 (EUF-RID-CMA): A RIBS scheme with outsourced revocation is said to be existentially unforgeable against adaptive chosen message and identity attacks if there is no probabilistic polynomial time adversary that has a non-negligible advantage in either Game I or Game II .

IV. PROPOSED RIBS SCHEME

In this section, we describe our proposed outsourced RIBS scheme. The scheme is composed of the following five algorithms, as defined in Section III-B.

- *Setup*(λ) : The KGC runs the algorithm as follows.
  1) Choose two cyclic groups G and G_1 with the same prime order q. Let P be a generator of group G, and $\hat{e} : G \times G \rightarrow G_1$ be a bilinear map. We compute $g = \hat{e}(P, P)$.
  2) We randomly choose two secret values $s, t \in Z_q^*$, where s is the master identity key and t is the maser time key. Then, we compute $P_{pub} = sP, P_t = tP$. Keep s secret and transform t to the CRS in a secure way.
  3) We select three hash functions as follows:
  $$H_1 : \{0,1\}^* \rightarrow G,$$
  $$H_2 : \{0,1\}^* \times \{0,1\}^* \rightarrow G,$$
  $$h : \{0,1\}^* \times G \rightarrow Z_q^*.$$
  4) We publish the system parameters
  $$PP = (q, G, G_1, P, P_{pub}, P_t, H_1, H_2, h).$$
- *InitKeyExt*(PP, s, ID) : For a user with identity ID, the KGC sets

$$Q_{ID} = H_1(ID), \quad D_{ID} = sQ_{ID},$$

and sends the initial identity key D_{ID} to the user through a secure channel.

- $TimeKeyUpd(PP, t, ID, T_i)$: Upon receiving an update request from a user ID at the time period T_i, the CRS computes

$$Q_{ID,T_i} = H_2(ID, T_i), \quad T_{ID,T_i} = tQ_{ID,T_i},$$

and sends T_{ID,T_i} to the user.

- $Sign(PP, m, ID, T_i, D_{ID}, T_{ID,T_i})$: Given a message m and time period T_i, a signer with identity ID produces the signature for m using the identity key D_{ID} and time update key T_{ID,T_i} as follows. We randomly choose $r \in Z_q^*$, and compute:

$$\begin{aligned} \alpha &= g^r, \\ v &= h(m, \alpha), \\ U &= rP + v(D_{ID} + T_{ID,T_i}). \end{aligned}$$

The signature for the message m at the time period T_i is $\sigma = (U, \alpha)$.

- $Ver(PP, m, \sigma, ID, T_i)$: On receiving a signature $\sigma = (U, \alpha)$ on message m and time period T_i, the verifier computes

$$v = h(m, \alpha),$$

and verifies if

$$\hat{e}(U, P) = \alpha \hat{e}(Q_{ID}, P_{pub})^v \hat{e}(Q_{ID,T_i}, P_t)^v.$$

holds. The verifier outputs "Accept" if it does, or "Reject" if not.

We demonstrate the consistency of the scheme as follows:

$$\begin{aligned} \hat{e}(U, P) &= \hat{e}(rP + v(D_{ID} + T_{ID,T_i}), P) \\ &= \hat{e}(rP, P)\hat{e}(D_{ID}, P)^v \hat{e}(T_{ID,T_i}, P)^v \\ &= \hat{e}(P, P)^r \hat{e}(sQ_{ID}, P)^v \hat{e}(tQ_{ID,T_i}, P)^v \\ &= g^r \hat{e}(Q_{ID}, sP)^v \hat{e}(Q_{ID,T_i}, tP)^v \\ &= \alpha \hat{e}(Q_{ID}, P_{pub})^v \hat{e}(Q_{ID,T_i}, P_t)^v. \end{aligned}$$

V. SECURITY ANALYSIS

In this section, we analyze the security of the proposed scheme in terms of the security model defined in section III-B. We employ the forking lemma technique introduced in [32].

Lemma 1: If there is a type I adversary who makes at most $q_{H_1}, q_{H_2}, q_h, , q_e, q_u, q_s$ queries to the hash functions H_1, H_2, h, initial key extract oracle, time key update oracle and signing oracle respectively and breaks the proposed RIBS scheme with non-negligible probability ϵ_I, then there exists a probabilistic challenger who can solve the CDH problem with advantage

$$\epsilon_I' \geq (1 - \frac{1}{q})\frac{1}{q_{H_2}}\epsilon_I - \frac{q_h}{q}.$$

Proof: Suppose A_I is a type I adversary who wins the attack game with advantage ϵ_I. We construct an algorithm S_I who uses A_I as a subroutine to solve the CDH problem. Suppose S_I is given a CDH instance $(P, P_a = aP, P_b = bP)$, where P is a generator of an additive cyclic group G of order q, and a, b is unknown to S_I. To compute $P_{ab} = abP$, S_I simulates a challenger for the adversary as follows.

- **Setup**. S_I randomly chooses $s \in Z_q^*$ and sets $P_{pub} = sP$, $P_t = P_a$ and sends (P, P_{pub}, P_t) to the adversary A_I. S_I chooses an $l \in \{1, 2, \ldots, q_{H_2}\}$ and maintains three lists L_1, L_2 and L_3 which are initially empty. S_I answers A_I's queries as follows.
- **Query**.
  - *Hash query*. We assume that A_I has already queried the corresponding hash oracles before it makes further queries. S_I answers three kinds of hash queries as follows.
    - * H_1*-query*. If A_I issues a H_1-query on identity ID, S_I first checks if there is an entry in the list L_1. If yes, S_I returns that entry, else S_I randomly chooses $x \in Z_q^*$ and returns $H_1(ID) = xP$ and adds $(ID, x, H_1(ID))$ into the list L_1.
    - * H_2*-query*. Suppose that A_I issues i-th H_2-query on identity ID_i and time period T_j, S_I first checks if there is an entry in the list L_2. If so, S_I returns that entry, else it randomly chooses $y \in Z_q^*$ and sets

$$H_2(ID_i, T_j) = \begin{cases} yP & i \neq l \\ yP_b & i = l \end{cases}$$

      S_I adds $(ID_i, T_j, y, H_2(ID_i, T_j))$ into list L_2 if $i \neq l$, else it adds the entry $(ID_i, T_j \perp, H_2(ID_i, T_j))$, and sets $ID^* = ID_i$ and $T^* = T_j$.
    - * h*-query*. On receiving a h-query on input (m, α), S_I first checks if there is an entry in the list L_3. If there is, S_I returns the entry, else it returns a randomly chosen $v \in Z_q^*$, and adds $(m, \alpha, h(m, \alpha))$ into list L_3.
  - *Initial key extract query*. On receiving such a query on identity ID, S_I searches list L_1 to find the entry $(ID, x, H_1(ID))$, and responds with $D_{ID} = xP_{pub}$.
  - *Time key update query*. If A_I issues a query on ID_i and T_j, S_I first checks if $(ID_i, T_j) = (ID^*, T^*)$. If not, S_I searches the list L_2 to find the entry $(ID_i, T_j, y, H_2(ID))$, and responds with $T_{ID_i,T_j} = yP_t$, else S_I sets $T_{ID_i,T_j} = \perp$.
  - *Signing query*. If A_I issues a signing query on identity ID_i, T_j and message m, S_I searches the list L_1, L_2, to find the corresponding $H_1(ID)$ and $H_2(ID, T_i)$. S_I then randomly chooses $U \in G$, $v \in Z_q^*$ and computes

$$\alpha = \hat{e}(U, P)\hat{e}(P_{pub}, H_1(ID_i))^{-v}\hat{e}(P_t, H_2(ID_i, T_j))^{-v}.$$

    S_I searches the list L_3, if there is an entry $(m, \alpha, h(m, \alpha))$ and $h(m, \alpha) \neq v$, then S_I aborts, else S_I returns the signature $\sigma = (U, \alpha)$ to A_I. In this case, σ is a valid signature.
- **Forgery**. Finally, the adversary A_I outputs a signature $\sigma^* = (U^*, \alpha^*)$ on ID', T', and message m^*.

If $(ID', T') = (ID^*, T^*)$ and $Ver(PP, m^*, ID^*, T^*) =$ "Accept", then output $\sigma^* = (U^*, \alpha^*)$. Otherwise, the output "fail" is issued.

Now we apply the forking lemma technique.

S_I runs the above simulated game again with the same random coins, but responds to hash queries issued by A_I with different random values. By the General Forking Lemma, A_I will output a different forgery $\sigma' = (U', \alpha')$ on the same message m^*, identity ID^* and time period T^* with non-negligible probability (the probability would be 1/9 for some appropriate chosen parameters. A more in-depth description is given in [32]. Here we assume that A_I always outputs another valid forgery without loss of generality). Since S_I runs the game with the same random tape, we have $\alpha^* = \alpha' = g^r$ from some $r \in Z_q^*$, while the underlying hash values corresponding to the two forged signatures are different. We assume that in the signature (U^*, α^*),

$$H_1(ID^*) = x^*P, \quad H_2(ID^*, T^*) = y^*P_b, \quad h(m^*, \alpha^*) = v^*.$$

While in the signature (U', α'),

$$H_1(ID^*) = x'P, \quad H_2(ID^*, T^*) = y'P_b, \quad h(m^*, \alpha') = v'.$$

Since the hash values are randomly chosen, so $x^* \neq x'$, $y^* \neq y'$ and $v^* \neq v'$ with high probability.

On the other hand, since both (U^*, α^*) and (U', α') are valid signatures, we have

$$\hat{e}(U^*, P) = \alpha^* \hat{e}(P_{pub}, x^*P)^{v^*} \hat{e}(P_t, y^*P_b)^{v^*} \quad (1)$$

$$\hat{e}(U', P) = \alpha' \hat{e}(P_{pub}, x'P)^{v'} \hat{e}(P_t, y'P_b)^{v'} \quad (2)$$

By dividing the above two equations, and the condition $\alpha^* = \alpha'$, we have

$$\begin{aligned}\hat{e}(U^* - U', P) &= \hat{e}(P_{pub}, P)^{x^*v^* - x'v'} \hat{e}(P_t, P_b)^{y^*v^* - y'v'} \\ &= \hat{e}(sP, P)^{x^*v^* - x'v'} \hat{e}(P_a, P_b)^{y^*v^* - y'v'} \\ &= \hat{e}(P, sP)^{x^*v^* - x'v'} \hat{e}(P, P_{ab})^{y^*v^* - y'v'},\end{aligned}$$

then

$$\begin{aligned}\hat{e}(P, P_{ab})^{y^*v^* - y'v'} &= \hat{e}(P, U^* - U')\hat{e}(P, sP)^{x'v' - x^*v^*} \\ &= \hat{e}(P, U^* - U')\hat{e}(P, (x'v' - x^*v^*)sP) \\ &= \hat{e}(P, (U^* - U') + (x'v' - x^*v^*)sP)\end{aligned}$$

So

$$\begin{aligned}\hat{e}(P, P_{ab}) &= \hat{e}(P, (U^* - U') + (x'v' - x^*v^*)sP)^{(y^*v^* - y'v')^{-1}} \\ &= \hat{e}(P, (y^*v^* - y'v')^{-1}(U^* - U' \\ &\quad + (x'v' - x^*v^*)sP))\end{aligned}$$

From the above equation we obtain:

$$P_{ab} = (y^*v^* - y'v')^{-1}(U^* - U' + (x'v' - x^*v^*)sP).$$

So we get the solution of the challenging CDH instance.

Now we analyze the probability that S_I succeeds. In the *Setup* and *Query* phase, the simulation is perfect except the following two events happen. First, S_I issues a query (ID^*, T^*) to H_2 oracle, which has a probability of q_{H_2}/q. Second, S_I returns a signature (U, α) on (m, ID_i, T_j) and $h(m, \alpha)$ has already been in the list L_3 and $h(m, \alpha) \neq v$, where v is randomly chosen by S_I. This event occurs with a probability of q_h/q. If the simulation process is executed smoothly, then in the *Forgery* phase, A_I will output a valid forgery (ID', T', m^*, σ^*) with an advantage of ϵ_I. Note that since H_2 is a random oracle, the probability that (ID', T', m^*, σ^*) is valid without any query of $H_2(ID', T')$ is $1/q$. So (ID', T') has been asked to H_2 oracle in the *Query* Phase with a probability of $1 - \frac{1}{q}$. Moreover, l is randomly chosen from $\{1, 2, \ldots, q_{H_2}\}$. Thus $(ID', T') = (ID^*, T^*)$ holds with a probability of $(1 - \frac{1}{q})\frac{1}{q_{H_2}}$. So the probability that $Ver(ID', T', m^*, \sigma^*) = accept$ and $(ID', T') = (ID^*, T^*)$ is $(1 - \frac{1}{q})\frac{1}{q_{H_2}}\epsilon_I - \frac{q_h}{q}$.

From the above analysis we can see that S_I solves the CDH problem with probability $(1 - \frac{1}{q})\frac{1}{q_{H_2}}\epsilon_I - \frac{q_h}{q}$.

Lemma 2: If there is a type II adversary who makes at most $q_{H_1}, q_{H_2}, q_h, , q_e, q_u, q_s$ queries to the hash functions H_1, H_2, h, initial key extract oracle, time key update oracle and signing oracle respectively and breaks the proposed RIBS scheme with a non-negligible probability ϵ_{II}, then there exists a probabilistic challenger who can solve the CDH problem with advantage

$$\epsilon'_{II} \geq (1 - \frac{1}{q})\frac{1}{q_{H_1}}\epsilon_{II} - \frac{q_h}{q}.$$

Proof: Suppose A_{II} is a type II adversary who wins the attack game with advantage ϵ_{II}. We construct an algorithm S_{II}, who uses A_{II} as a subroutine to solve the CDH problem. Suppose S_{II} is given a CDH instance $(P, P_a = aP, P_b = bP)$, where P is a generator of an additive cyclic group G of order q, and a, b is unknown to S_{II}. To compute $P_{ab} = abP$, S_{II} simulates a challenger for the adversary as follows.

- **Setup**. S_{II} randomly choose $t \in Z_q^*$, and sets $P_{pub} = P_a$, $P_t = tP$ and sends (P, P_{pub}, P_t) to the adversary A_{II}. S_{II} then randomly chooses $l \in \{1, 2, \ldots, q_{H_1}\}$. S_{II} maintains three lists L_1, L_2 and L_3 which are initially empty, and answers A_{II}'s queries as follows.
- **Query**.
  - *Hash query*. We assume that the adversary has already queried the corresponding hash oracles before it makes further queries.
    - H_1-*query*. On receiving the i-th H_1-query on identity ID_i, S_{II} first checks if there is an entry in the list L_1. If there is, S_{II} returns the entry, else S_{II} randomly chooses $x \in Z_q^*$ and sets

$$H_1(ID_i) = \begin{cases} xP & i \neq l \\ xP_b & i = l \end{cases}$$

S_{II} returns $H_1(ID_i)$ to A_{II} and adds $(ID_i, x, H_1(ID_i))$ into the list L_1 if $i \neq l$, otherwise it adds $(ID_i, \perp, H_1(ID_i))$ into list L_1 and set $ID^* = ID_i$.
    - H_2-*query*. On receiving a H_2-query on ID_i and T_j, S_{II} first checks if there is an entry in the list L_2. If yes, S_{II} returns as the same, else it randomly chooses $y \in Z_q^*$ and returns

$H_2(ID_i, T_j) = yP$. S_{II} then adds $(ID_i, T_j, y, H_2(ID_i, T_j))$ into list L_2.

* *h-query*. On receiving a h-query on input (m, α), S_{II} first checks if there is an entry in the list L_3. If yes, S_{II} returns as the same, otherwise it returns a randomly chosen $v \in Z_q^*$, and adds $(m, \alpha, h(m, \alpha))$ into list L_3.

– *Initial key extract query*. If A_{II} issues such a query on identity ID, S_{II} first checks if $ID = ID^*$. If not, S_{II} searches list L_1 to find the entry $(ID, x, H_1(ID))$, and responds with $D_{ID} = xP_{pub}$, else S_{II} returns a $\perp$.

– *Time key update query*. On receiving such a query on ID_i and T_j, S_{II} searches the list L_2 to find the entry $(ID_i, T_j, y, H_2(ID_i, T_j))$, and responds with $T_{ID_i,T_j} = yP_t$.

– *Signing query*. If A_{II} issues a signing query on ID, T_i and message m, S_{II} first searches the list L_1, L_2, L_3 to find the corresponding $H_1(ID)$ and $H_2(ID, T_i)$. S_{II} then randomly chooses $U \in G, v \in Z_q^*$ and computes

$$\alpha = \hat{e}(U, P)\hat{e}(P_{pub}, H_1(ID))^{-v}\hat{e}(P_t, H_2(ID, T_i))^{-v}.$$

S_{II} searches the list L_3, if there is an entry $(m, \alpha, h(m, \alpha))$ and $h(m, \alpha) \neq v$, then S_{II} aborts, else S_{II} returns the signature $\sigma = (U, \alpha)$ to A_{II}. In this case, σ is a valid signature.

- **Forgery**. The adversary A_{II} outputs a forgery $\sigma^* = (U^*, \alpha^*)$ on identity ID', time period T', and message m^*. If $ID' = ID^*$ and $Ver(PP, m^*, ID', T') = accept$, S_{II} outputs (U^*, α^*) as the forgery else it outputs "fail."

S_{II} runs the simulated game twice with the same random coins, but responds the hash queries with different random values. By the General Forking Lemma, A_{II} will output a different forgery $\sigma' = (U', \alpha')$ on the same identity ID^*, message m^* and time period T_i^* with non-negligible probability. We have $\alpha^* = \alpha' = g^r$, and assume that in the signature (U^*, α^*),

$$H_1(ID^*) = x^*P_b, \quad H_2(ID^*, T_i^*) = y^*P_t, \quad h(m^*, \alpha^*) = v^*.$$

While in the signature (U', α'),

$$H_1(ID^*) = x'P_b, \quad H_2(ID^*, T_i^*) = y'P_t, \quad h(m^*, \alpha^*) = v'.$$

Since the hash values are randomly chosen, so $x^* \neq x'$, $y^* \neq y'$ and $v^* \neq v'$ with overwhelming probability.

On the other hand, since both (U^*, α^*) and (U', α') are valid signatures, we have

$$\hat{e}(U^*, P) = \alpha^*\hat{e}(P_{pub}, x^*P_b)^{v^*}\hat{e}(P_t, y^*P)^{v^*} \quad (3)$$
$$\hat{e}(U', P) = \alpha'\hat{e}(P_{pub}, x'P_b)^{v'}\hat{e}(P_t, y'P)^{v'} \quad (4)$$

By dividing the above two equations, and the condition $\alpha^* = \alpha'$, we have

$$\begin{aligned}\hat{e}(U^* - U', P) &= \hat{e}(P_{pub}, P_b)^{x^*v^* - x'v'}\hat{e}(P_t, P)^{y^*v^* - y'v'}\\ &= \hat{e}(aP, bP)^{x^*v^* - x'v'}\hat{e}(tP, P)^{y^*v^* - y'v'}\\ &= \hat{e}(P, P_{ab})^{x^*v^* - x'v'}\hat{e}(P, tP)^{y^*v^* - y'v'},\end{aligned}$$

TABLE 2. Notations of computation costs.

Notation	Description
$TG_{\hat{e}}$:	Time cost of a bilinear pairing map operation.
TG_m :	Time cost of a scalar multiplication operation.
TG_a :	Time cost of a point addition operation.
TG_H :	Time cost of a map-to-point hash function operation.
T_e :	Time cost of a modular exponentiation operation in G_T.
T_m :	Time cost of a multiplication operation in G_T.
T_h :	Time cost of an ordinary hash function operation.

TABLE 3. Computation time for operations.

Operation	Time (ms)
$TG_{\hat{e}}$:	5.275
TG_H :	5.101
TG_m :	1.970
T_e :	0.331
T_h :	0.009
TG_a :	0.003
T_m :	0.001

then

$$\begin{aligned}\hat{e}(P, P_{ab})^{x^*v^* - x'v'} &= \hat{e}(P, U^* - U')\hat{e}(P, tP)^{y'v' - y^*v^*}\\ &= \hat{e}(P, U^* - U')\hat{e}(P, (y'v' - y^*v^*)tP)\\ &= \hat{e}(P, (U^* - U') + (y'v' - y^*v^*)tP).\end{aligned}$$

So

$$\begin{aligned}\hat{e}(P, P_{ab}) &= \hat{e}(P, (U^* - U')\\ &\quad + (y'v' - y^*v^*)tP)^{(x^*v^* - x'v')^{-1}}\\ &= \hat{e}(P, (x^*v^* - x'v')^{-1}((U^* - U')\\ &\quad + (y'v' - y^*v^*)tP))\end{aligned}$$

From the above equation we can see that

$$P_{ab} = (x^*v^* - x'v')^{-1}((U^* - U') + (y'v' - y^*v^*)tP).$$

The analysis of S_{II}'s advantage is just the same as S_I's advantage in the simulated Game I. We note that at the end of simulated Game II, A_{II} will output a valid signature on identity ID^* with a probability of $(1 - \frac{1}{q})\frac{1}{q_{H_1}}\epsilon_{II} - \frac{q_h}{q}$, which is exactly the advantage that S_{II} succeeds. This concludes the proof.

From Lemma 1 and Lemma 2 we get the following theorem.

Theorem 1: The proposed RIBS scheme with outsourced revocation is existence unforgeable against adaptive chosen identity and message attack under the CDH assumption.

VI. PERFORMANCE EVALUATION

In this section we present the performance evaluation of the proposed scheme, including computation and communication costs. We choose the Ate pairing $\hat{e} : G \times G \rightarrow G_T$ generated by a point on a super singular eppliptic curve

TABLE 4. Comparisons of computation costs.

Schemes	Initial Key Extraction	Time Key Update	Signing	Verifying
Hess's IBS scheme[12]	$TG_m + TG_H$	–	$TG_{\hat{e}} + 2TG_m + T_e$	$2TG_{\hat{e}} + T_e$
cost(ms)	7.071	–	9.546	10.881
Paterson's IBS scheme[10]	$TG_m + TG_H$	–	$3TG_m$	$2TG_{\hat{e}} + 2T_e$
cost(ms)	7.071	–	5.91	11.212
Tsai *et al.*'s RIBS scheme[28]	$3TG_m$	$3TG_m$	$4TG_m$	$4TG_{\hat{e}}$
cost(ms)	5.91	5.91	7.88	21.1
Hung *et al.*'s RIBS scheme[29]	$3TG_m$	$3TG_m$	$5TG_m$	$4TG_{\hat{e}} + T_e$
cost(ms)	5.91	5.91	9.85	21.431
Our proposed scheme	$TG_m + TG_H$	$TG_m + TG_H$	$2TG_m$	$TG_{\hat{e}} + 2T_e$
cost(ms)	7.071	7.071	3.94	5.937

over a finite field $E(F_p)$, where G, G_T are groups of prime order q. To ensure an appropriate security level, p and q are large prime numbers with a length of 512 and 160 bits respectively. Table 2 lists the notations used to describe the computation costs of the operations used in the related schemes.

Previous implementations have showed that compared with the computation costs of time-consuming bilinear pairing map, map-to-point hash, scalar multiplication and modular exponentiation operations, the computation costs of point addition, multiplication in G_T and the ordinary hash operations are trivial. Therefore, we only consider $TG_{\hat{e}}, TG_H, TG_m, T_e$ when we evaluate the performance.

We evaluate the costs of the above basic operations using MIRACL library [33] on the elastic compute service (ECS) host provided by the Alibaba Cloud platform. The operating system of the host is Ubutu 14.04 for 64 bit with an Intel(R) Xeon(R) CPU E5-2630 0 @ 2.30GHz, and equipped with 1GB RAM. Table 3 lists the computational time for related operations on the host.

Since there is no other RIBS scheme with CRS based on pairings and proven secure in the random oracle model, we therefore compared the performance of our scheme with some IBS schemes with bilinear pairings but without revocation functionality, and we also compared the performance of our scheme with two RIBS schemes that are secure in the standard model. Table 4 lists the comparisons among the schemes of Hess [12], Paterson [10], Tsai *et al.* [28], Hung *et al.* [29] and ours in terms of computation costs for the initial key extraction, time key update, signing and verification.

There is no revocation in Hess's and Paterson's schemes, but we can extend their schemes to be revocable by using the framework we proposed in section III-A. Tsai and Hung have included revocation functionality in their schemes, but the schemes are designed in the standard model, which means that the hash functions they used are much more inefficient than those used in the random oracle model. In the comparisons, although we omit most of the computation costs of specific, hash operations, the comparison results are still meaningful and referential. We also take into account the operations which can be precomputed in all the schemes to achieve the best performance from them.

For the computation cost in the initial key extraction, Hess's scheme requires $TG_m + TG_H$ (7.071ms), same as Paterson's and our scheme. Tsai's and Hung's schemes both require $3TG_m$ (5.91ms). As for the time key update, our scheme requires $TG_m + TG_H$ (7.071ms). Tsai's and Hung's schemes require $3TG_m$ (5.91ms). There is no such operations in Hess's and Paterson's scheme, but if we extend their schemes using the technique we propose, the computation costs of time key update of their schemes will be same as ours, namely, $TG_m + TG_H$ (7.071). As for the signing process, Hess's scheme requires $TG_{\hat{e}} + 2TG_m + T_e$ (9.546ms). Paterson's scheme requires $3TG_m$ (5.91ms). Tsai's scheme requires $4TG_m$ (7.88ms). Hung's scheme requires $5TG_m$ (9.85ms) while our scheme only requires $2TG_m$ (3.94ms). For the verification process, although there are three bilinear maps that need to be evaluated for each signature in our scheme, but some of them can be precomputed. Hess's scheme requires $2TG_{\hat{e}} + T_e$ (10.881ms). Paterson'sS scheme requires $2TG_{\hat{e}} + 2T_e$ (11.212ms). Tsai's scheme requires $4TG_{\hat{e}}$(21.1ms). Hung's scheme requires $4TG_{\hat{e}} + T_e$ (21.431ms). Our scheme requires $TG_{\hat{e}} + 2T_e$ (5.937ms). Although our scheme seems a little more time consuming than Tsai's and Hung's schemes in the initial key extraction and time key update process, it is worth pointing out that there are several point addition brevity. Moreover, the initial key extraction and time key update process would not be executed frequently, so there is impact on the overall performance. As for the signing and verification process, our scheme outperforms the other schemes.

Table 5 presents the comparisons of communication costs in terms of the size of initial identity key, time update key and signature. Let $|G|$ denote the size of each element in group G. If G is a elliptic curve on finite field F_p, where p is a 512 bit prime number, then $|G|$ denotes the size of a point in G, which is 1024 bits. $|q|$ denotes the bit length of q, which is, for example, 160 bits to achieve an appropriate security level. The initial identity key has a length of $|G|$ (1024 bits) in schemes of Hess, Paterson and ours, and has a length of $2|G|$ (2048 bits) in the schemes of Tsai *et al.* and Hung *et al.*.

TABLE 5. Comparison of communication costs.

Schemes	Size of Initial Key	Size of Time Key	Size of Signature	Security Model	Revocability
Hess's IBS scheme[12]	$\|G\|$	–	$\|G\|+\|q\|$	RO	No
length (bits)	1024	–	1184		
Paterson's IBS scheme[10]	$\|G\|$	–	$2\|G\|$	RO	No
length (bits)	1024	–	2048		
Tsai *et al.*'s RIBS scheme[28]	$2\|G\|$	$2\|G\|$	$4\|G\|$	STD	Yes
length (bits)	2048	2048	4096		
Hung *et al.*'s RIBS scheme[29]	$2\|G\|$	$2\|G\|$	$4\|G\|$	STD	Yes
length (bits)	2048	2048	4096		
Our proposed scheme	$\|G\|$	$\|G\|$	$\|G\|+\|q\|$	RO	Yes
length (bits)	1024	1024	1184		

The time update key has the same length as the initial identity key in each scheme except for Hess's and Paterson's schemes. For the size of the signature, Hess's scheme has a length of $|G|+|q|$ (1184 bits), which is same as ours. Paterson's scheme has a length of $2|G|$ (2048 bits). Both Tsai's and Hung's schemes has a length of $4|G|$ (4096 bits). We observe that the communication costs of Hess's scheme and ours are lower than the other schemes. We also present the security model and revocability of the target schemes in table 5.

VII. CONCLUSION

In this paper, we propose an efficient RIBS scheme with CRS based on bilinear pairings. To eliminate the computation and communication costs of the KGC, revocation functionality is outsourced to a cloud revocation server. We present the framework of the outsourced revocation RIBS scheme and formalize the security model. Our scheme is proven to be secure against existential forgery on adaptively chosen messages and identity attacks in the random oracle model. The performance comparisons show that our scheme has lower computation costs and shorter signature size than previously proposed RIBS schemes thereby demonstrating its suitability for resource-constrained resources such as wireless sensor networks.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their constructive comments which helped us improve the quality and presentation of this paper.

REFERENCES

[1] D. Boneh and M. Franklin, "Identity-based encryption from the weil pairing," in *Proc. Annu. Int. Cryptol. Conf.*, 2001, pp. 213–229.

[2] J. Li, J. Li, X. Chen, C. Jia, and W. Lou, "Identity-based encryption with outsourced revocation in cloud computing," *IEEE Trans. Comput.*, vol. 64, no. 2, pp. 425–437, Feb. 2015.

[3] A. Shamir, "Identity-based cryptosystems and signature schemes," in *Proc. Workshop Theory Appl. Cryptograph. Techn.*, 1984, pp. 47–53.

[4] A. Fiat and A. Shamir, "How to prove yourself: Practical solutions to identification and signature problems," in *Proc. Adv. Cryptol.*, 1999, pp. 420–426.

[5] L. C. Guillou and J. J. Quisquater, *A 'Paradoxical' Indentity-Based Signature Scheme Resulting from Zero-Knowledge*, New York, NY USA: Springer, 1990, pp. 216–231.

[6] T. Okamoto, *Provably Secure and Practical Identification Schemes and Corresponding Signature Schemes*. Berlin, Germany: Springer, 1992.

[7] H. Tanaka, "A realization scheme for the identity-based cryptosystem," *Electron. Commun. Jpn.*, vol. 73, no. 5, pp. 340–349, 1990.

[8] R. Sakai, "Cryptosystems based on pairing," in *Proc. Symp. Cryptogr. Inf. Secur. (SCIS)*, Jan. 2001, pp. 26–28.

[9] J. C. Choon and J. H. Cheon, "An identity-based signature from gap diffie-hellman groups," in *Proc. Int. Workshop Theory Pract. Public Key Cryptogr., Public Key*, 2002, pp. 18–30.

[10] K. G. Paterson, "ID-based signatures from pairings on elliptic curves," *Electron. Lett.*, vol. 38, no. 18, pp. 1025–1026, Aug. 2002.

[11] F. Hess, "Exponent group signature schemes and efficient identity based signature schemes based on pairings," 2002.

[12] F. Hess, *Efficient Identity Based Signature Schemes Based on Pairings*. Berlin, Germany: Springer, 2003.

[13] D. Galindo and F. D. Garcia, "A schnorr-like lightweight identity-based signature scheme," in *Proc. 2nd Int. Conf. Cryptol. Africa Prog. Cryptol.-AFRICACRYPT*, Gammarth, Tunisia, Jun. 2009, pp. 135–148.

[14] M. Bellare, C. Namprempre, and G. Neven, "Security proofs for identity-based identification and signature schemes," *J. Cryptol.*, vol. 22, no. 1, pp. 1–61, 2009.

[15] F. Zhang, R. Safavi-Naini, and W. Susilo, "An efficient signature scheme from bilinear parings and its applications," in *Proc. Int. Workshop Theory Pract. Public Key Cryptogr.*, Singapore, Mar. 2004, pp. 277–290.

[16] P. S. L. M. Barreto, B. Libert, N. Mccullagh, and J. J. Quisquater, *Efficient and Provably-Secure Identity-Based Signatures and Signcryption From Bilinear Maps*. Berlin, Germany: Springer, 2005.

[17] B. Waters, *Efficient Identity-Based Encryption Without Random Oracles*. Berlin, Germany: Springer, 2005.

[18] K. G. Paterson and J. C. N. Schuldt, *Efficient Identity-Based Signatures Secure in the Standard Model*. Berlin, Germany: Springer, 2006.

[19] F. Zhang and K. Kim, *Efficient ID-Based Blind Signature and Proxy Signature From Bilinear Pairings*. Berlin, Germany: Springer, 2003.

[20] X. Chen, F. Zhang, D. M. Konidala, and K. Kim, "A new id-based group signature scheme from bilinear pairings," in *Proc. Int. Workshop Inf. Secur. Appl.*, vol. 3348. 2003, pp. 585–592.

[21] J. H. Cheon, Y. Kim, and H. J. Yoon, "A new id-based signature with batch verification," in *Proc. Cryptol. Eprint Arch. (IACR)*, 2004, p. 131.

[22] G. U. Wei-Na, "A new id-based group signature scheme," *Comput. Modernization*, 2010.

[23] A. Boldyreva, V. Goyal, and V. Kumar, "Identity-based encryption with efficient revocation," in *Proc. ACM Conf. Comput. Commun. Secur. (CCS)*, Alexandria, VA, USA, Oct. 2008, pp. 417–426.

[24] B. Libert and J.-J. Quisquater, "Efficient revocation and threshold pairing based cryptosystems," in *Proc. 22nd Annu. Symp. Principles Distrib. Comput.*, 2003, pp. 163–171.

[25] J. H. Seo and K. Emura, "Revocable identity-based encryption revisited: Security model and construction," in *Proc. Public-Key Cryptogr.-PKC*, 2013, pp. 216–234.

[26] J. H. Seoa and K. Emurab, "Revocable hierarchical identity-based encryption," *Theoretical Comput. Sci.*, vol. 542, pp. 44–62, Jul. 2014.

[27] Y.-M. Tseng and T.-T. Tsai, "Efficient revocable ID-based encryption with a public channel," *Comput. J.*, vol. 55, no. 4, pp. 475–486, 2012.

[28] T. T. Tsai, Y. M. Tseng, and T. Y. Wu, "Provably secure revocable id-based signature in the standard model," *Secur. Commun. Netw.*, vol. 6, no. 10, pp. 1250–1260, 2013.
[29] Y. H. Hung, T. T. Tsai, Y. M. Tseng, and S. S. Huang, "Strongly secure revocable id-based signature without random oracles," *Inf. Technol. Control*, vol. 43, no. 3, pp. 264–276, 2014.
[30] Y. Sun, F. Zhang, L. Shen, and R. Deng, "Revocable identity-based signature without pairing," in *Proc. Int. Conf. Intell. Netw. Collaborat. Syst.*, 2013, pp. 363–365.
[31] J. Wei, W. Liu, and X. Hu, "Forward-secure identity-based signature with efficient revocation," *Int. J. Comput. Math.*, vol. 93, pp. 1–23, 2016.
[32] D. Pointcheval and J. Stern, "Security arguments for digital signatures and blind signatures," *J. Cryptol.*, vol. 13, no. 3, pp. 361–396, 2000.
[33] M. Scott, "Miracl library," (2011). [Online] Available: http://www.shamus.

XIAOYING JIA received the B.S. and M.S. degrees from PLA Information Engineering University in 1999 and 2004, respectively, and the Ph.D. degree from the State Key Laboratory of Information Security, Graduate University of Chinese Academy of Sciences, in 2012. She is currently a Lecturer with South-Central University for Nationalities. Her research interests include applied cryptography, cloud computing, and network security.

DEBIAO HE received the Ph.D. degree in applied mathematics from the School of Mathematics and Statistics, Wuhan University, in 2009. He is currently an Associate Professor with the State Key Laboratory of Software Engineering, Computer School, Wuhan University. His main research interests include cryptography and information security, in particular, cryptographic protocols.

SHERALI ZEADALLY received a bachelor's degree from the University of Cambridge, United Kingdom, and his doctorate degree from the University of Buckingham, United Kingdom, both in computer science. He is an associate professor in the College of Communication and Information, University of Kentucky, Lexington, USA. He is a Fellow of the British Computer Society and a Fellow of the Institution of Engineering Technology, United Kingdom.

LI LI received the Ph.D. degree in computer science from the School of Computer, Wuhan University, in 2004. She is currently an Associate Professor with the International School of Software, Wuhan University. Her research interests lie in the area of security and privacy, including privacy protection, embedded security, and data security.