IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received September 6, 2018, accepted September 21, 2018, date of publication October 1, 2018,
date of current version October 25, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2873019

CCA-Secure Revocable Identity-Based Encryption With Ciphertext Evolution in the Cloud

YINXIA SUN[1], WILLY SUSILO[2], (Senior Member, IEEE), FUTAI ZHANG[1], AND ANMIN FU[3], (Member, IEEE)

[1]School of Computer Science and Technology, Nanjing Normal University, Nanjing 210023, China
[2]School of Computing and Information Technology, University of Wollongong, Wollongong, NSW 2500, Australia
[3]School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing 210094, China

Corresponding author: Yinxia Sun (bela_suno@163.com)

This work was supported by the Nature Science Foundation of China under Grant 61502237, Grant 61672289, and Grant 61572255.

ABSTRACT Identity-based encryption (IBE) is a very attractive cryptographic primitive due to its unnecessity of any certificate managements. Nevertheless, the user revocation problem in IBE remains an elusive research problem and hence, it is an important research topic. One possible approach in achieving revocations is to update user's decryption keys. However, to avoid the need of secret channels, public time keys need to be issued to allow this update to occur. It is unfortunate that this method often suffers from two problems: 1) the user has to maintain linearly growing decryption keys; and 2) the revoked users can still access ciphertexts prior to revocation. At the first glance, proxy re-encryption technique may provide a solution to this problem, but the ciphertexts will become longer after each re-encryption, which makes it impractical. In this paper, we present a revocable identity-based encryption scheme with cloud-aided ciphertext evolution. Our construction solves the two aforementioned problems via ciphertext evolution implemented by the cloud. In addition, the size of ciphertexts in the cloud remains constant size regardless of evolutions. The scheme is provably secure against chosen ciphertext attacks based on the BDH problem. The comparisons with the existing related works show that our scheme enjoys better efficiency, and thus it is practical for the data sharing in cloud storage.

INDEX TERMS CCA, ciphertext evolution, cloud, identity-based encryption, revocable.

I. INTRODUCTION

Public key encryption (PKE) provides an excellent solution to the problem of key distribution in symmetric key. An important issue in PKE is the authenticity of user public keys. The traditional PKE authenticates user public key via releasing certificates. Nevertheless, the certificate management is a heavy burden to the public key system, which is the primary drawback in PKE. To overcome this drawback, Shamir set forth a new notion called ''Identity based public key cryptography'' in 1984 [16]. This public key cryptosystem employs every user's unique identity as its public key. Therefore, the authenticity of this public key is no longer questionable, and hence, there is no certificate required. Since the first practical identity-based encryption (IBE) scheme was presented by Boneh and Franklin [3] in 2001, IBE has attracted a lot of attentions from both academia and industry. To date, there have been many IBE schemes proposed in the literature such as [2] and [4]–[7]. One important issue to make identity-based encryption practical is the user revocation. This problem was first discussed in BF's seminal work [3]. As stated earlier, different from the traditional public key system, there is no certificate in identity-based system. Therefore, the conventional user revocation technique is not applicable to the identity-based systems. Actually, the user private key can be viewed as an implicit certificate. Boneh and Franklin suggested that the PKG periodically issues new private keys by attaching time tags for non-revoked users. The user revocation can be launched by the PKG stoping the issuing of new user private keys with the time tags. Unfortunately, this revocation system is very impractical, because the PKG has to carry heavy overhead ($O(n)$ where n is the number of non-revoked users), especially for the establishing of secret channels. Boldyreva *et al.* [1] presented the first scalable revocable IBE scheme in 2008. In their scheme, only public channels are required for the key-updating. Furthermore, they utilized the complete subtree to realize a logarithm growth $O(\log(n))$ of key-updating with non-revoked users.

With this approach of revocation method, many revocable identity-based encryption (RIBE) schemes have been proposed. However, when taking these schemes in some

application scenarios, some problems arise. Let's consider the scenario of secure data sharing in cloud storage by applying a revocable identity-based encryption. Suppose there are four entities involved, namely the data owner, the data user, the cloud server and the PKG. To our best knowledge, the existing schemes suffer from two shortcomings or at least one of them.

- The data user needs to utilize the time key $\mathsf{TK}_{ID,t}$ as well as the private key to decrypt a ciphertext encrypted at the time t. So, the data user has to maintain all the time keys ($O(t)$) (or decryption keys computed from time keys) for different time-period decryptions. This consumes a lot of storage resources for data users, thus it is very impractical especially in source-limited environments.
- When a user is revoked by the PKG for such as private key compromise or expiring, the user can still decrypt those ciphertexts prior to revocation in the cloud.

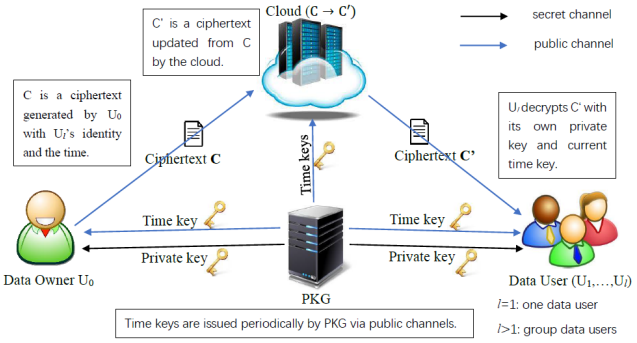

FIGURE 1. System model of RIBE with ciphertext evolution in the cloud.

Though some works [10] do not have these problems, they suffer from new problems: the ciphertexts grow longer with the number of ciphertext-transformation (the basic construction), or the costly computation and communication of re-encryption keys between the user and the server. We give a new and efficient RIBE scheme that can solve the two problems. As shown in Fig. 1, the cloud server helps a ciphertext C uploaded by the data owner evolve to a new one C'; then the data user decrypts the ciphertext by employing merely its current time key as well as private key. So,

- ⋆ the data user only needs to keep one time key – the current time key;
- ⋆ and the revoked data user cannot access any ciphertext including those before revocation;
- ⋆ no matter how many times a ciphertext in the cloud evolves, the length of the ciphertext remains unchanged;
- ⋆ different to proxy re-encryption, no computation or communication is needed for the re-encryption keys.

We construct a revocable identity-based encryption scheme with (cloud-aided) ciphertext evolution (RIBE-CE). Our scheme is proved secure against chosen ciphertext attacks based on the hardness of the BDH problem in the random oracle model. We also analyze the efficiency by comparing our scheme with the existing works.

Organization of the Paper: The rest of the paper is organized as follows. In Sect. 2, we give the definition of a cloud-aided RIBE-CE scheme and its security model against two types of adversaries. The related mathematical notions are also reviewed in this section. Sect. 3 presents the concrete RIBE-CE construction and the formal security proofs. In Sect. 4, we analyze the efficiency by comparing with other related representative works. Finally, Sect. 5 concludes the paper.

A. RELATED WORKS

Boldyreva *et al.* [1] first presented a scalable revocable IBE scheme in 2008. The complete subtree structure is used to obtain a logarithm growth of updating key. This work was subsequently improved in [12] with strong security. In 2013, Seo and Emura introduced decryption-key-exposure threat and presented a revocable IBE scheme that can resist decryption-key-exposure [15]. Sun *et al.* [17] extended Seo *et al.*'s scheme to the certificateless setting. In 2017, Watanabe *et al.* [20] gave a new revocable and decryption-key-exposure resistent IBE scheme. The scheme has short public parameters in prime-order groups. A very recent work showed a revocable hierarchical identity-based encryption scheme [8].

Unfortunately, almost all these RIBE works suffer from the two drawbacks mentioned above especially in the applications like cloud storage. In 2014, Liang *et al.* [10] employed proxy re-encryption technique to construct a revocable identity-based encryption scheme. Their scheme solves the two problems by re-encrypting ciphertexts in the cloud. However, the ciphertexts become longer and longer with the number of re-encryption. Reference [14] improved Liang *et al.*'s scheme on the efficiency but suffers from increasing list of decryption keys for different-period ciphertexts. References [18] and [19] pointed out the security weakness of Liang *et al.*'s scheme against collusion attacks. Other related works are such as [9] and [13]. In these works, the revocation is implemented by a third party e.g. the cloud server. They have special applications.

II. DEFINITION AND SECURITY MODEL

A. SCHEME DEFINITION

An RIBE-CE scheme is made up of the following algorithms.

- **Setup**(k): Taking a security parameter k as input, this algorithm outputs a master secret key `msk` and public parameters `params`.
- **Private-Key-Extract**(`params`, `msk`, ID): Taking `params`, `msk` and an identity ID as input, this algorithm outputs a private key SK_{ID}, which is transmitted to the user via a secret channel. It is run by the PKG
- **Time-Key-Update**(`params`, `msk`, ID, t): Taking `params`, `msk`, an identity ID and a time tag t as input, this algorithm outputs a time key $\mathsf{TK}_{ID,t}$, which is transmitted to the user via a public channel. It is run by the PKG

- **Encrypt**(params, ID, t, M): Taking params, ID, t and a message M as input, this algorithm outputs a ciphertext C. It is run by the data owner.
- **Decrypt**(params, SK_{ID}, $\mathsf{TK}_{ID,t}$): Taking params, SK_{ID}, $\mathsf{TK}_{ID,t}$ and C as input, this algorithm outputs a message M or a failure symbol. It is run by the data user.
- **Revoke**(ID, t): Taking ID and t as input, the PKG stops issuing the time key $\mathsf{TK}_{ID,t}$ for the user.
- **Ciphertext-Evolve**(params, C, $\mathsf{TK}_{ID,t}$, t'): Taking $\mathsf{TK}_{ID,t}$, t' and C as input, this algorithm outputs a ciphertext C'. It is run by the cloud server.

B. SECURITY MODEL

Two types of adversaries are considered: an outside adversary who knows all time keys; an inside adversary who is a malicious revoked user. The cloud server can be viewed as an outside adversary.

We define the IND-CCA security of revocable identity based encryption with ciphertext evolution via the following game between the challenger $\mathcal{C}$ and the adversary $\mathcal{A}$. Let $\mathcal{A}$ be $\mathcal{A}_o$ or $\mathcal{A}_i$ that denote an outside adversary or an inside adversary, respectively.

Setup: $\mathcal{C}$ runs the setup algorithm to provide public parameters params to $\mathcal{A}$, while keeps the master secret key msk for itself.

Phase 1: Then, $\mathcal{A}$ may make some private key queries $\mathcal{Q}_{\mathsf{privatekey}}$, time key queries $\mathcal{Q}_{\mathsf{timekey}}$, decryption queries $\mathcal{Q}_{\mathsf{decryption}}$ and ciphertext evolution queries $\mathcal{Q}_{\mathsf{ciphertextevolution}}$ to the challenger $\mathcal{C}$ on its behalf. The query-answers are described as follows.

$\mathcal{Q}_{\mathsf{privatekey}}$	$\mathcal{A} \xrightarrow{ID} \mathcal{C}, \mathcal{C} \xrightarrow{\mathsf{SK}_{ID}} \mathcal{A}$
$\mathcal{Q}_{\mathsf{timekey}}$	$\mathcal{A} \xrightarrow{(ID,t)} \mathcal{C}, \mathcal{C} \xrightarrow{\mathsf{TK}_{ID,t}} \mathcal{A}$
$\mathcal{Q}_{\mathsf{decryption}}$	$\mathcal{A} \xrightarrow{(C,ID,t)} \mathcal{C}, \mathcal{C} \xrightarrow{M} \mathcal{A}$
$\mathcal{Q}_{\mathsf{ciphertextevolution}}$	$\mathcal{A} \xrightarrow{C_{ID,t},t'} \mathcal{C}, \mathcal{C} \xrightarrow{C_{ID,t'}} \mathcal{A}$

Challenge: $\mathcal{A}$ outputs two messages M_0 and M_1 of the same length, an identity ID^* and a time tag t^*. The challenger randomly chooses β from $\{0, 1\}$ and encrypts M_β to output a challenge ciphertext C^*.

Phase 2: $\mathcal{A}$ continues to make queries as before, subject to the constrain that $\mathcal{A}_o$ cannot make a private key extraction query on ID^* and $\mathcal{A}_i$ cannot make a time key query on (ID^*, t^*).

Guess: Finally, $\mathcal{A}$ outputs a guess $\beta' \in \{0, 1\}$.

The advantage of $\mathcal{A}$ in the above game is defined by $\epsilon = |\Pr(\beta' = \beta) - 1/2|$. An RIBE-CE scheme is said to be IND-CPA (indistinguishability against chosen plaintext attacks) secure if no PPT adversary has non-negligible ϵ.

C. BILINEAR PAIRING AND COMPLEXITY ASSUMPTION

Bilinear paring. $\mathbb{G}_1$ is an additive group with prime order q and a generator P. $\mathbb{G}_2$ is a multiplicative group with the same order q. A *bilinear pairing* is defined as $e : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$ satisfying the conditions (1) as described below.

$$\begin{cases} \textit{Bilinearity} : a, b \in \mathbb{Z}_q^*, e(aP, bP) = e(P, P)^{ab}; \\ \textit{Non} - \textit{degeneracy} : e(P, P) \neq 1_{\mathbb{G}_2}; \\ \textit{Computability}: e(aP,bP) \textit{ can be effectively computed}. \end{cases} \quad (1)$$

Bilinear Diffie-Hellman (BDH) problem. *Given* $(aP, bP, cP \in \mathbb{G}_1)$ *with* $a, b, c \in_R \mathbb{Z}_q^*$, *to compute* $e(P, P)^{abc}$.

III. THE CONSTRUCTION

A. GENERIC CONSTRUCTION

Generally suppose an identity based encryption scheme is made up of four algorithms $\mathsf{IBE}^{\mathsf{CCA}}_{\mathsf{Setup}}$, $\mathsf{IBE}^{\mathsf{CCA}}_{\mathsf{Extract}}$, $\mathsf{IBE}^{\mathsf{CCA}}_{\mathsf{Enc}}$ and $\mathsf{IBE}^{\mathsf{CCA}}_{\mathsf{Dec}}$. We can give a generic construction of revocable identity-based encryption with (cloud-aided) ciphertext evolution (RIBE-CE for short).

– **Setup**(k):
$\mathsf{IBE}^{\mathsf{CCA}}_{\mathsf{Setup}}(k) \to (\mathsf{params}, \mathsf{msk})$.

– **Private-Key-Extract**(params, msk, ID):
$\mathsf{IBE}^{\mathsf{CCA}}_{\mathsf{Extract}}(\mathsf{params}, \mathsf{msk}, ID) \to \mathsf{SK}_{ID}$.

–**Time-Key-Update**(params, msk, ID, t):
$\mathsf{IBE}^{\mathsf{CCA}}_{\mathsf{Extract}}(\mathsf{params}, \mathsf{msk}, ID||t) \to \mathsf{TK}_{ID,t}$;
$\mathsf{TK}_{ID,t}$ is sent to both the user and the cloud server.

–**Encrypt**(params, ID, t, M):
$C_0 = \mathsf{IBE}^{\mathsf{CCA}}_{\mathsf{Enc}}(M, ID)$;
$C = \mathsf{IBE}^{\mathsf{CCA}}_{\mathsf{Enc}}(C_0, ID||t)$;
output the final ciphertext C.

–**Decrypt**(params, SK_{ID}, $\mathsf{TK}_{ID,t}$, C):
$C_0 = \mathsf{IBE}^{\mathsf{CCA}}_{\mathsf{Dec}}(C, \mathsf{TK}_{ID,t})$;
$M = \mathsf{IBE}^{\mathsf{CCA}}_{\mathsf{Dec}}(C_0, \mathsf{SK}_{ID})$.

–**Revoke**(ID, t):
The PKG does not generat the time key $\mathsf{TK}_{ID,t}$.

–**Ciphertext-Evolve**(params, C, $\mathsf{TK}_{ID,t}$, t'):
$C_0 = \mathsf{IBE}^{\mathsf{CCA}}_{\mathsf{Dec}}(C, \mathsf{TK}_{ID,t})$;
$C' = \mathsf{IBE}^{\mathsf{CCA}}_{\mathsf{Enc}}(C_0, ID||t')$;
$C \xrightarrow{Evolve} C'$.

The security of the generic construction can be provided by Theorem 1 below.

Theorem 1: If the underlying IBE scheme is IND-CCA secure then our generic construction of RIBE-CE is IND-CCA secure against both an outside adversary $\mathcal{A}_o$ and an inside adversary $\mathcal{A}_i$.

Proof: Suppose $\mathcal{B}$ is an adversary against the IBE scheme. It will acts as the challenger interacting with $\mathcal{A}$. $\mathcal{B}$ has a list of IBE public parameters params which is published to $\mathcal{A}$ as well as some other necessary parameters. Then $\mathcal{A}$ may make some queries.

Private key extraction query:

$$\mathcal{A} \xrightarrow{ID} \mathcal{B} \xrightarrow{ID} \mathcal{C},$$
$$\mathcal{C} \xrightarrow{\mathsf{SK}_{ID}} \mathcal{B} \xrightarrow{\mathsf{SK}_{ID}} \mathcal{A}.$$

Time key query:

$$\mathcal{A} \xrightarrow{ID,t} \mathcal{B} \xrightarrow{ID||t} \mathcal{C},$$
$$\mathcal{C} \xrightarrow{\mathsf{SK}_{ID||t}} \mathcal{B} \xrightarrow{\mathsf{SK}_{ID||t}} \mathcal{A},$$
$$\mathsf{TK}_{ID,t} = \mathsf{SK}_{ID||t}.$$

Ciphertext evolution query: For $\mathcal{A} \xrightarrow{(C,ID,t,t')} \mathcal{B}$, $\mathcal{B}$ firstly searches the time key list for $\mathsf{TK}_{ID,t}$ and $\mathsf{TK}_{ID,t'}$, then computes $C_0 = \mathsf{IBE}_{\mathsf{Dec}}(C, \mathsf{TK}_{ID,t})$ and $C' = \mathsf{IBE}_{\mathsf{Enc}}(C_0, ID||t')$. At last, $\mathcal{C} \xrightarrow{C'} \mathcal{A}$.

If ID has been revoked at time t', $\mathsf{TK}_{ID,t'}$ is randomly chosen by the cloud.

Decryption query: For $\mathcal{A} \xrightarrow{(C,ID,t)} \mathcal{C}$, firstly $\mathcal{B}$ uses $\mathsf{TK}_{ID,t}$ to make a decryption as $C_0 = \mathsf{IBE}^{\mathsf{CCA}}_{\mathsf{Dec}}(C, \mathsf{TK}_{ID,t})$. Then

$$\mathcal{B} \xrightarrow{C_0,ID} \mathcal{C},$$
$$\mathcal{C} \xrightarrow{M} \mathcal{B},$$
$$\mathcal{B} \xrightarrow{M} \mathcal{A}.$$

Challenge: $\mathcal{A}$ selects two messages (M_0, M_1), an identity ID^* and a time t^* as the challenge. $\mathcal{B}$ sends (M_0, M_1, ID^*) to its challenger then receives a challenge ciphertext C_0^*. $\mathcal{B}$ further computes $C^* = \mathsf{IBE}^{\mathsf{CCA}}_{\mathsf{Enc}}(C_0^*, ID^*||t^*)$ by inputting $(C_0^*, ID^*||t^*)$ as a message and an identity. Return C^* to $\mathcal{A}$ as the challenge ciphertext.

A may continue to make queries as before, subject to the following constrains

- the private key extraction query on ID^* is not allowed; (especially for $\mathcal{A}_o$)
- the time key query on (ID^*, t^*) is not allowed; (especially for $\mathcal{A}_i$)
- the decryption query on C^* of (ID^*, t^*) is not allowed.

Guess. In the end, $\mathcal{A}$ gives a guess $\beta \in \{0, 1\}$. $\mathcal{B}$ output the same guess.

If $\mathcal{A}$'s advantage to win the above game is ϵ, it is clear that $\mathcal{B}$'s advantage to break the IND-CCA security of the IBE scheme is not less than ϵ.

B. CONCRETE CONSTRUCTION

To be concrete, the construction can be efficient.

- **Setup**(k): Select two cyclic groups $(\mathbb{G}_1, +)$ and $(\mathbb{G}_2, \cdot)$ of the same order q. P is a generator of $\mathbb{G}_1$. $e : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$ is a bilinear pairing. Choose $s \in \mathbb{Z}_q^*$ at random and compute $P_0 = sP$. Select four hash functions: $H_1 : \{0,1\}^* \to \mathbb{G}_1$, $H_2 : \{0,1\}^* \to \mathbb{G}_1$, $H_3 : \{0,1\}^l \times \{0,1\}^l \to \mathbb{Z}_q^*$, $H_4 : \mathbb{G}_2 \to \{0,1\}^l$ and $H_5 : \{0,1\}^l \to \{0,1\}^l$. l is the message length.
  The system parameters `params` are
  $$\langle \mathbb{G}_1, \mathbb{G}_2, q, P, P_0, e, H_1, \ldots, H_5 \rangle.$$
  The master secret key `mk` is s.
- **Private-Key-Extract**(`params`, `mk`, ID): The PKG computes $Q_{ID} = H_1(ID)$ and then calculates $\mathsf{SK}_{ID} = sQ_{ID}$ as the private key of user ID.
- **Time-Key-Update**(`params`, `mk`, ID, t): The PKG computes $Q_{ID,t} = H_2(ID, t)$ and then calculates $\mathsf{TK}_{ID,t} = sQ_{ID,t}$ as the time key of the user ID at the time t.
- **Encrypt**(`params`, M, ID, t): The data owner does the following to encrypt a message M.
  - choose $\sigma \in \{0,1\}^l$, compute $r = H_3(\sigma, M)$;
  - compute $U = rP$;
  - compute
    $$V = \sigma \oplus H_4(e(Q_{ID}, P_0)^r) \oplus H_4(e(Q_{ID,t}, P_0)^r);$$
  - compute
    $$W = M \oplus H_5(\sigma);$$
  - output the ciphertext $C = (U, V, W)$.
- **Decrypt**(`params`, $C = (U, V, W)$, SK_{ID}, $\mathsf{TK}_{ID,t}$): The data user computes
  $$\sigma = V \oplus H_4(e(\mathsf{SK}_{ID}, U)) \oplus H_4(e(\mathsf{TK}_{ID,t}, U)).$$
  Then it can recover the message by calculating
  $$M = W \oplus H_5(\sigma).$$
  If the equation $U = H_3(\sigma, M)P$ holds, the message M is correct.
- **Revoke**(ID, t): If the user ID with identity needs to be revoked at the time t, the PKG stops generating the time key $\mathsf{TK}_{ID,t}$ for the user.
- **Ciphertext-Evolve**(`params`, $C = (U, V, W)$, ID, $\mathsf{TK}_{ID,t}$, $\mathsf{TK}_{ID,t'}$):
  The cloud computes
  $$V' = V \oplus H_4(e(\mathsf{TK}_{ID,t}, U)) \oplus H_4(e(\mathsf{TK}_{ID,t'}, U));$$
  then updates the ciphertext to be $C' = (U, V', W)$.

Correctness 1. To recover the plaintext from the ciphertext $C = (U, V, W)$, the key point is to compute σ. So, we first verify the correctness of σ.

$$\begin{aligned}\sigma &= V \oplus H_4(e(\mathsf{SK}_{ID}, U)) \oplus H_4(e(\mathsf{TK}_{ID,t}, U))\\ &= \sigma \oplus H_4(e(Q_{ID}, P_0)^r) \oplus H_4(e(Q_{ID,t}, P_0)^r)\\ &\quad \oplus H_4(e(sQ_{ID}, rP)) \oplus H_4(e(sQ_{ID,t}, rP))\\ &= \sigma \oplus H_4(e(sQ_{ID}, rP)) \oplus H_4(e(sQ_{ID,t}, rP))\\ &\quad \oplus H_4(e(sQ_{ID}, rP)) \oplus H_4(e(sQ_{ID,t}, rP))\\ &= \sigma.\end{aligned}$$

Then

$$\begin{aligned}M &= W \oplus H_5(\sigma)\\ &= M \oplus H_5(\sigma) \oplus H_5(\sigma)\\ &= M.\end{aligned}$$

So, if the equation $U = H_3(\sigma, M)P$ holds, the message M is correct.

Correctness 2. To verify the decryption on the ciphertext after evolution $C' = (U, V', W)$, we can observe that

$$\begin{aligned} V' &= V \oplus H_4(e(\mathsf{TK}_{ID,t}, U)) \oplus H_4(e(\mathsf{TK}_{ID,t'}, U)) \\ &= \sigma \oplus H_4(e(Q_{ID}, P_0)^r) \oplus H_4(e(Q_{ID,t}, P_0)^r) \\ &\quad \oplus H_4(e(\mathsf{TK}_{ID,t}, U)) \oplus H_4(e(\mathsf{TK}_{ID,t'}, U)) \\ &= \sigma \oplus H_4(e(Q_{ID}, P_0)^r) \oplus H_4(e(\mathsf{TK}_{ID,t}, U)) \\ &\quad \oplus H_4(e(\mathsf{TK}_{ID,t}, U)) \oplus H_4(e(\mathsf{TK}_{ID,t'}, U)) \\ &= \sigma \oplus H_4(e(Q_{ID}, P_0)^r) \oplus H_4(e(\mathsf{TK}_{ID,t'}, U)) \\ &= \sigma \oplus H_4(e(Q_{ID}, P_0)^r) \oplus H_4(e(Q_{ID,t'}, P_0)^r). \end{aligned}$$

Obviously, the V' in $C' = (U, V', W)$ keeps the encryption algorithm. So the decryption is also suitable to the evolving ciphertexts.

Remark: If the data user is revoked, the cloud randomly chooses $\mathsf{TK}_{ID,t'} \in \mathbb{G}_1$ for the ciphertext evolution. From the point of practical view, 1) the revocation is also the cloud server's wish; 2) the message is for the data user himself but not for other persons. It is different from proxy re-encryption. So, if the revoked user and the cloud server collude, though they can access the data encrypted before revocation, we can ignore this kind of attack.

C. THE SECURITY

Theorem 2: The hash functions are viewed as random oracles. If there exists an outside adversary $\mathcal{A}_o$ against the IND-CCA security of our scheme with advantage ϵ, making q_1 times H_1 hash queries and q_4 times H_4 hash queries, then there exists an algorithm $\mathcal{B}$ that can solve the BDH problem with probability $\epsilon' \geq \epsilon/(q_1 \cdot q_4)$.

Proof: Suppose $\mathcal{B}$ intends to solve the BDH problem with random instances $< aP, bp, cP >$ and its ultimate goal $e(P, P)^{abc}$. Now it will act the challenger to interact with the adversary $\mathcal{A}_o$. In the beginning, $\mathcal{B}$ setups public parameters as $< \mathbb{G}_1, \mathbb{G}_2, q, P, P_0 = aP, H_i(1 \leq i \leq 5) >$.

Before $\mathcal{A}_o$ makes some queries, it selects $I \in [1, q_1]$.

Hash queries. All hash queries are answered by randomly choosing a proper element. The detailed query-answers are $< (ID_i, x_i \in_R \mathbb{Z}_q^*, x_iP) >$ for H_1, $< (ID_i, , t_i, y_i \in_R \mathbb{Z}_q^*, y_iP) >$ for H_2, $< (\sigma, M, r) >$ for H_3, $< (E, h) >$ for H_4, $< (\sigma, \tilde{h}) >$ for H_5. Especially, when $i = I$, $\mathcal{B}$ responds with $H_1(ID_i) = bP$.

Private key queries. When $\mathcal{B}$ receives a private key query on (ID_i), it searches the H_1 list for $< (ID_i, x_i, x_iP) >$ and computes $SK_{ID_i} = x_iaP$ as the answer. Note that if $i = I$, $\mathcal{B}$ aborts the game.

Time key queries. When $\mathcal{B}$ receives a time key query on (ID_i, t_i), it searches the H_2 list for $< (ID_i, t_i, y_i, y_iP) >$ and computes $TK_{ID_i,t_i} = y_iaP$ as the answer.

Decryption queries. When $\mathcal{B}$ receives a decryption query on $(C = (U, V, W), ID, t)$, it runs **Decrypt**$(\mathtt{params}, C, \mathsf{SK}_{ID}, \mathsf{TK}_{ID,t})$ by extracting the private key SK_{ID} and the time key $\mathsf{TK}_{ID,t}$ firstly. If $i = I$, without loss of indistinguishability from the reality, suppose the ciphertext is not generated by $\mathcal{A}_o$ itself and $\mathcal{C}$ acts as follows.

- Choose $M \in_R \{0, 1\}^l$ and return M as the answer;
- Select $h \in_R \{0, 1\}^l$ Set $H_4(\mathsf{BDH}(Q_{ID}, U, P_0)) = h$;
- Compute $\sigma = V \oplus h \oplus H_4(TK_{ID,t}, U)$ and set $H_5(\sigma) = W \oplus M$;
- Pick $r \in_R \mathbb{Z}_q^*$ and set $H_3(\sigma, M) = r$.

Actually, due to the difficulty of Discrete Logarithm ($U = \tilde{r}P \rightarrow \tilde{r}$), the equation $r = \tilde{r}$ holds just with negligible probability. So, if $\mathcal{A}_o$ makes the $H_3(\sigma, M)$ query, the answer is easy to be found incorrect. However, it is not difficult to see that if the $H_3(\sigma, M)$ query occurs, $\mathcal{A}_o$ has made $H_4(\mathsf{BDH}(Q_{ID_i}, U, P_0))$ before with non-negligible probability for the computation of σ. Since $\mathsf{BDH}(Q_{ID_i}, U, P_0)$ is a BDH problem, our decryption query implementation seems real to the adversary $\mathcal{A}_o$.

After that, $\mathcal{A}_o$ outputs two messages (M_0, M_1) and the identity ID^* and the time period t^* that it wants to challenge. If $ID^* \neq ID_I$, $\mathcal{B}$ aborts the game; otherwise, $\mathcal{B}$ generates a challenge ciphertext $C^* = (U^*, V^*, W^*)$ by setting

$$U^* = cP, \quad V^* \in_R \{0, 1\}^l, \quad W^* \in_R \{0, 1\}^l.$$

$\mathcal{A}_o$ continues to make more queries as before, except the private key query on ID^* and the decryption query on (C^*, ID^*, t^*).

At the end of the game, $\mathcal{A}_o$ outputs its guess for β. $\mathcal{B}$ picks a tuple (E, h) at random from the H_4 list and outputs the E as the solution to the BDH problem.

Analysis. If $\mathcal{B}$ does not abort the game and $\mathcal{A}_o$ can break the IND-CCA security of the scheme with advantage ϵ, $\mathcal{B}$ can solve the BDH problem with probability $\epsilon' \geq \epsilon/q_4$. Clearly, the game will not abort if $ID^* = ID_I$. So, $\epsilon' \geq \epsilon/(q_1 \cdot q_4)$.

Theorem 3: The hash functions are viewed as random oracles. If there exists an inside adversary $\mathcal{A}_i$ against the IND-CCA security of our scheme with advantage ϵ, making q_2 times H_2 hash queries and q_4 times H_4 hash queries, then there exists an algorithm $\mathcal{B}$ that can solve the BDH problem with probability $\epsilon' \geq \epsilon/(q_2 \cdot q_4)$.

Proof: In this proof, $\mathcal{B}$ is still a solver to the BDH problem with random instances $< aP, bp, cP >$ and its goal is to compute $e(P, P)^{abc}$. It acts as the challenger interacting with the adversary $\mathcal{A}_i$. $\mathcal{B}$ setups public parameters as $< \mathbb{G}_1, \mathbb{G}_2, q, P, P_0 = aP, H_i(1 \leq i \leq 5) >$.

Before $\mathcal{A}_i$ makes queries, it selects $I \in [1, q_2]$ and suppose the Ith query to the H_2 oracle is on (ID^*, t^*).

Hash queries. All hash queries are answered by randomly choosing a proper element. The detailed query-answers are $< (ID_i, x_i \in_R \mathbb{Z}_q^*, x_iP) >$ for H_1, $< (ID_i, , t_j, y \in_R \mathbb{Z}_q^*, yP) >$ for H_2, $< (\sigma, M, r) >$ for H_3, $< (E, h) >$ for H_4, $< (\sigma, \tilde{h}) >$ for H_5. Especially, when $(ID_i, t_j) = (ID^*, t^*)$, $\mathcal{B}$ responds with $H_2(ID_i, t_j) = bP$.

Private key queries. When $\mathcal{B}$ receives a private key query on (ID_i), it searches the H_1 list for $< (ID_i, x_i, x_iP) >$ and computes $SK_{ID_i} = x_iaP$ as the answer.

Time key queries. When $\mathcal{B}$ receives a time key query on (ID_i, t_j), it searches the H_2 list for $< (ID_i, t_j, y, yP) >$ and computes $TK_{ID_i,t_j} = yaP$ as the answer. Note that if $(ID_i, t_j) = (ID^*, t^*)$, $\mathcal{B}$ aborts the game.

Decryption queries. When $\mathcal{B}$ receives a decryption query on $(C = (U, V, W), ID, t)$, it runs **Decrypt**(params, C, SK_{ID}, $\mathsf{TK}_{ID,t}$) by extracting the private key SK_{ID} and the time key $\mathsf{TK}_{ID,t}$ firstly. If $(ID_i, t_j) = (ID^*, t^*)$, without loss of indistinguishability from the reality, suppose the ciphertext is not generated by $\mathcal{A}_i$ itself and $\mathcal{C}$ works as follows.

- Choose $M \in_R \{0,1\}^l$ and return M as the answer;
- Select $h \in_R \{0,1\}^l$ Set $H_4(\mathsf{BDH}(Q_{ID,t}, U, P_0)) = h$;
- Compute $\sigma = V \oplus h \oplus H_4(SK_{ID}, U)$ and set $H_5(\sigma) = W \oplus M$;
- Pick $r \in_R \mathbb{Z}_q^*$ and set $H_3(\sigma, M) = r$.

Similarly, due to the difficulty of Discrete Logarithm ($U = \tilde{r}P \rightarrow \tilde{r}$), the equation $r = \tilde{r}$ holds just with negligible probability. So, if $\mathcal{A}_i$ makes the $H_3(\sigma, M)$ query, the answer is easy to be found incorrect. However, it is not difficult to see that if the $H_3(\sigma, M)$ query occurs, $\mathcal{A}_i$ has made $H_4(\mathsf{BDH}(Q_{ID,t}, U, P_0))$ before with non-negligible probability for the computation of σ. Since to compute $\mathsf{BDH}(Q_{ID,t}, U, P_0)$ is a hard problem, our decryption simulation seems real to the adversary $\mathcal{A}_i$.

After that, $\mathcal{A}_i$ launches the challenge by outputting two messages (M_0, M_1) and an identity ID^* and a time period t^*. $\mathcal{B}$ generates a challenge ciphertext $C^* = (U^*, V^*, W^*)$ by setting

$$U^* = cP, \quad V^* \in_R \{0,1\}^l, \quad W^* \in_R \{0,1\}^l.$$

$\mathcal{A}_i$ continues to make more queries as before, except the time key query on (ID^*, t^*) and the decryption query on (C^*, ID^*, t^*).

At the end of the game, $\mathcal{A}_i$ outputs its guess for β. $\mathcal{B}$ picks a tuple (E, h) at random from the H_4 list and outputs the E as the solution to the BDH problem.

Analysis. Similar to the analysis of the above proof, If $\mathcal{B}$ does not abort the game and $\mathcal{A}_i$ can break the IND-CCA security of the scheme with advantage ϵ, $\mathcal{B}$ can solve the BDH problem with probability $\epsilon' \geq \epsilon/q_4$. Since the probability that the game does not abort is $1/q_2$, the probability $\epsilon' \geq \epsilon/(q_2 \cdot q_4)$.

D. THE EFFICIENCY

This section evaluates the efficiency by comparing our scheme with some representative related schemes [15], [10] and [14]. We build the experiment platform on a windows 10 machine which is equiped with Intel(R) Core(TM) i5-4460S CPU colcked at 2.9GHZ and 4GB system memory. The cryptography library that we choose is JPBC Library. The element length of group $\mathbb{G}_1$ and $\mathbb{G}_2$ in our scheme is 512-bit. For a 128-bit message, the running time for **Private-Key-Extract**, **Time-Key-Update**, **Encrypt**, **Decrypt** and **Ciphertext-Evolve** is 13ms, 13ms, 33ms, 16ms and 8ms, respectively. Table 1 makes efficiency comparison via main calculations, the exponential computation and the bilinear pairing. The algorithms include 1) generations of private key, time key, decryption key and re-encryption key; 2) encryption, decryption and ciphertext evolution. Then we make further comparison in Table 2, including a) the length of the decryption key list and the ciphertext after evolution; b) whether the ciphertexts prior to revocation are still available to the revoked user; c) the security level and the party implementing revocation. Table 3 describes the notations.

TABLE 1. Comparison of computation.

Algorithm	[15]	[10]	[14]	Ours
Private key	$2\tilde{e}$	$2\tilde{e}$	$2\tilde{e}$	$\tilde{e}$
Time key	$3\tilde{e}$	$3\tilde{e}$	$3\tilde{e}$	$\tilde{e}$
Decryption key	$4\tilde{e}$	$7\tilde{e}$	$5\tilde{e}$	-
Re-encryption key	-	$6\tilde{e}$	$5\tilde{e}$	-
Encryption	$4\tilde{e}+p$	$5\tilde{e}+p$	$5\tilde{e}+p$	$3\tilde{e}+2p$
Decryption	$3p$	$O(l)(\tilde{e}+p)$	$3p$	$2p$
Ciphertext evolution	-	$2p$	$3p$	$2p$

TABLE 2. Comparison.

Scheme	DK-list-size	CT-size	Q2?	Security	rev-by
[15]	$O(t)$	-	yes	CPA	PKG
[14]	$O(t)$	$O(1)$	yes	CPA	Server
[10]	$O(1)$	$O(l)$	no	CPA	Sever
Ours	$O(1)$	$O(1)$	no	CCA	PKG

TABLE 3. Notations.

Notation	Description
$\tilde{e}$	the exponential computation
p	the pairing computation
l	the number of ciphertext evolutions
t	the number of the time peoriods
CT-size	the size of ciphertext after evolutions
DK-list-size	the size of user decryption key
Q2	decryptions available on ciphertexts prior to revocation
rev-by	revocation is implemented by the PKG or the Server

From the comparisons above, except for the comparable efficiency to the existing schemes in encryption and decryption, we can see that our scheme performs better in the following aspects:

- more efficient in generations of private key, time key, decryption key and re-encryption key, and decryption;
- solving the two problems simultaneously with constant ciphertext after evolutions in the cloud.

IV. CONCLUSION

This paper focussed on the user revocation problem in identity-based encryption. One of the efficient revocation methods is to issue time keys periodically via public channels for non-revoked users. We take the scenario of cloud storage as consideration and find that most of the existing works suffer from increasing decryption key list or accessability to ciphertexts prior to revocation. Though the technique of proxy re-encryption can solve the two problems, the length of the ciphertexts grow linearly with the number of ciphertext-evolutions. Therefore, it is a heavy burden to the server when the data is big. Additionally, the users have to put more computation and communication resource on re-encryption keys. This is not suitable for source-limited applications. In this paper, we presented an efficient solution to the two aforementioned problems simultaneously. The size of a ciphertext

in the cloud remains constant, no matter how many times the ciphertext evolves. The new revocable identity-based encryption with ciphertext evolution scheme is constructed by using bilinear parings. The time keys are generated by the PKG periodically and sent to both the user and the cloud. The cloud makes ciphertext evolution by using the time keys. Hence, no extra key computations are involved in our construction. It is efficient for the data sharing application in the cloud. Our scheme enjoys provably strong security against chosen ciphertext attacks based on standard hard problem.

REFERENCES

[1] A. Boldyreva, V. Goyal, and V. Kumar, "Identity-based encryption with efficient revocation," in *Proc. CCS*, 2008, pp. 417–426.
[2] D. Boneh and X. Boyen, "Efficient selective-ID secure identity-based encryption without random oracles," in *Proc. EUROCRYPT*, in Lecture Notes in Computer Science, vol. 3027, 2004, pp. 223–238.
[3] D. Boneh and M. Franklin, "Identity-based encryption from the weil pairing," in *Proc. CRYPTO*, in Lecture Notes in Computer Science, vol. 2139, 2001, pp. 213–229.
[4] D. Boneh and X. Boyen, "Secure identity based encryption without random oracles," in *Proc. CRYPTO*, in Lecture Notes in Computer Science, vol. 3152, 2004, pp. 443–459.
[5] R. Brent, "Efficient identity-based encryption without random oracles," in *Proc. EUROCRYPT*, in Lecture Notes in Computer Science, vol. 3494, 2005, pp. 114–127.
[6] C. Cocks, "An identity based encryption scheme based on quadratic residues," in *Cryptography and Coding* (Lecture Notes in Computer Science), vol. 2260, B. Honary, Eds. Berlin, Germany: Springer, 2001.
[7] C. Gentry, "Practical identity-based encryption without random oracles," in *Proc. EUROCRYPT*, in Lecture Notes in Computer Science, vol. 4004, 2006, pp. 445–464.
[8] K. Lee and S. Park, "Revocable hierarchical identity-based encryption with shorter private keys and update keys," *Des., Codes Cryptogr.*, vol. 86, no. 10, pp. 2407–2440, 2018.
[9] 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.
[10] K. Liang, J. K. Liu, D. S. Wong, and W. Susilo, "An efficient cloud-based revocable identity-based proxy re-encryption scheme for public clouds data sharing," in *Proc. 19th Eur. Symp. Res. Comput. Secur.*, vol. 8712. Sep. 2014, pp. 257–272.
[11] B. Libert and J.-J. Quisquater, "Efficient revocation and threshold pairing based cryptosystems," in *Proc. PODC*, 2003, pp. 163–171.
[12] B. Libert and D. Vergnaud, "Adaptive-ID secure revocable identity-based encryption," in *Proc. CT-RSA*, in Lecture Notes in Computer Science, vol. 5473, 2009, pp. 1–15.
[13] K. Nguyen, H. Wang, and J. Zhang, "Server-aided revocable identity-based encryption from lattices," in *Proc. CANS*, in Lecture Notes in Computer Science, vol. 10052, 2016, pp. 107–123.
[14] B. Qin, R. H. Deng, Y. Li, and S. Liu, "Server-aided revocable identity-based encryption," in *Proc. ESORICS*, in Lecture Notes in Computer Science, vol. 9326, 2015, pp. 286–304.
[15] J. H. Seo and K. Emura, "Revocable identity-based encryption revisited: Security model and construction," in *Proc. PKC*, in Lecture Notes in Computer Science, vol. 7778, 2013, pp. 216–234.
[16] A. Shamir, "Identity-based cryptosystems and signature schemes," in *Proc. CRYPTO*, 1984, pp. 47–53.
[17] Y. Sun, F. Zhang, L. Shen, and R. H. Deng, "Efficient revocable certificate-less encryption against decryption key exposure," *IET Inf. Secur.*, vol. 9, no. 3, pp. 158–166, 2015.
[18] Y.-M. Tseng, T.-T. Tsai, S.-S. Huang, and C.-P. Huang, "Identity-based encryption with cloud revocation authority and its applications," *IEEE Trans. Cloud Comput.*, to be published, doi: 10.1109/TCC.2016.2541138.
[19] C. Wang, Y. Li, J. Fang, and J. Xie, "Cloud-aided scalable revocable identity-based encryption scheme with ciphertext update," *Concurrency Comput., Pract. Exper.*, vol. 29, no. 20, p. e4035, 2017.
[20] Y. Watanabe, K. Emura, and J. H. Seo, "New revocable IBE in prime-order groups: Adaptively secure, decryption key exposure resistant, and with short public parameters," in *Proc. CT-RSA*, in Lecture Notes in Computer Science, vol. 10159, 2017, pp. 432–449.

YINXIA SUN received the bachelor's and master's degrees in mathematics from Nanjing Normal University, China, and the D.Phil. degrees from Xidian University, China. She is currently an Associate Professor at Nanjing Normal University. Her main research interests include cryptography and cloud storage. She received the title of Excellent Member from CACR in 2015.

WILLY SUSILO received the bachelor's degree *(Summa Cum Laude)* predicate in computer science from Universitas Surabaya, Indonesia, and the master's and D.Phil. degrees from UOW. He is the Director of the Institute of Cybersecurity and Cryptology. His main research interests include cryptography and cyber security. He received the prestigious ARC Future Fellowship from the Australian Research Council. He also received the UOW Researcher of the Year 2016 due to his research excellence.

FUTAI ZHANG received the bachelor's and master's degrees in mathematics from Shanxi Normal University, China, and the D.Phil. degrees from Xidian University, China. He is currently a Professor at Nanjing Normal University. His main research interests include cryptography and applications of cryptography in cyberspace security.

ANMIN FU received the Ph.D. degree in information security from Xidian University in 2011. He is currently an Associate Professor and also the Supervisor of Ph.D. students of Nanjing University of Science and Technology, China. His research interests include cloud computing security, IoT security, and privacy preserving. He has published over 50 technical papers, including international journals and conferences, such as the IEEE TRANSACTIONS ON BIG DATA, the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, the IEEE COMMUNICATIONS LETTERS, the *Journal of Network and Computer Applications*, *Computers & Security*, and *Cluster Computing*.