Sun *et al. Cybersecurity* (2020) 3:12
https://doi.org/10.1186/s42400-020-00054-6

Cybersecurity

RESEARCH Open Access

Server-aided immediate and robust user revocation mechanism for SM9

Shuzhou Sun[1,2], Hui Ma[1*], Rui Zhang[1,2] and Wenhan Xu[1,2]

Abstract

As the only approved Identity-Based Encryption scheme in China that is also standardized by ISO, SM9-IBE has been widely adopted in many real-world applications. However, similar to other IBE standard algorithms, SM9-IBE currently lacks revocation mechanism, which is vital for a real system. Worse still, we find that existing revocable techniques may not be suitable and efficient when applying to SM9-IBE. Given the widespread use of SM9-IBE, an efficient and robust user revocation mechanism becomes an urgent issue.

In this work, we propose a dedicated server-aided revocation mechanism, which for the first time achieves the secure, immediate and robust user revocation for SM9-IBE. Provided with a compact system model, the proposed method leverages an existing server to perform all heavy workloads during user revocation, thus leaving no communication and computation costs for the key generation center and users. Moreover, the mechanism supports key-exposure resistance, meaning the user revocation mechanism is robust even if the revocation key leaks. We then formally define and prove the security. At last, we present theoretical comparisons and an implementation in terms of computational latency and throughput. The results indicate the efficiency and practicability of the proposed mechanism.

Keywords: Identity-based encryption, SM9, Server-aided immediate and robust revocation, Chinese cryptography standard, Security proof, Performance evaluation

Introduction

Identity-Based Encryption (IBE) is a special kind of public key encryption, where a user utilizes a unique string (e.g., an email address or a phone number) as the public key. Data senders do not have to obtain receivers' public key certificates, thus eliminating the Public Key Infrastructure (PKI). In the past decades, IBE has been thoroughly studied and many practical IBE schemes have been proposed, e.g., (Boneh and Franklin 2001; Cocks 2001; Canetti et al. 2004; Boneh and Boyen 2004a; 2004b; Waters 2005). Further, some of the proposed schemes were standardized in a number of globally recognized standards (Boyen and Martin 2007; Martin et al. 2009; Martin and Schertler 2009; IEEE 2013; Iso/iec 2015).

SM9 (Gm/t 2016a) is a Chinese national cryptography standard for Identity-Based Cryptography, which consists of 3 cryptographic primitives: a digital signature scheme, a key agreement scheme and an encryption scheme. The SM9 encryption scheme (denoted as SM9-IBE) is also standardized in ISO 18033-5 (Iso/iec 2015). After its standardization, SM9-IBE has been extensively applied in many scenarios, including encrypted emails, electronic government systems and commercial products.

Though SM9-IBE is proposed as a standard algorithm, it does not specify any user revocation mechanisms in the standards (Iso/iec 2015) and (Gm/t 2016a). However, since SM9-IBE is practice-oriented, it is desirable to deal with the realistic problem of user revocation. On the other hand, although a large body of the previous work (Boldyreva et al. 2008; Libert and Vergnaud 2009; Seo and Emura 2013; Li et al. 2013; Qin et al. 2015; Ge and Wei 2019) has been done for efficient IBE user revocation, all these known techniques are not generic, namely, they cannot be applied to SM9-IBE. The major technical

*Correspondence: mahui@iie.ac.cn
[1]State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, China
Full list of author information is available at the end of the article

challenge is that SM9-IBE has a different mathematical structure compared with the existing revocable IBE schemes, which were construed with certain IBE schemes and relied on their concrete mathematical structures, thus adopting these solutions will face with the difficulties in both the construction and the security proof.

As a result, the problem of efficient SM9-IBE user revocation is still open. As a national and international standard for IBE, it is desirable to have a practical user revocation mechanism for SM9-IBE.

Our Contributions. Aiming at solving the revocation problem of SM9-IBE, we propose a dedicated Server-Aided Immediate and Robust Revocable Identity-Based Encryption scheme (denoted as SA-IR-RIBE), which can be effortlessly integrated into the existing systems. Remarkably, our methodology achieves direct user revocation by performing simple operations in the server. Our results lie in the following aspects:

- **A Compact System with Revocation.** In this paper, we present a practical revocation mechanism for SM9-IBE. To the best of our knowledge, this is the first specific solution to the revocation problem for SM9-IBE. By introducing a helping server, the proposed mechanism has a compact system model and does not introduce additional entities. The heavy workloads of user revocation are all performed by the server, leaving no communication and computation costs for both the key generation center (KGC) and users.
- **Efficient and Immediate Revocation.** Unlike most of the existing revocable IBE schemes, our scheme enjoys instant revocation, in the sense that a user cannot decrypt ciphertexts at the same moment when he is revoked. In our scheme, all ciphertexts have to be partially decrypted by the server before it is decrypted by a user. Therefore, a user can be immediately revoked from the system by taking simple operations on the server side.
- **Server Side Key-Exposure Resistance.** In an ordinary scheme, once the server secret key is revealed, the revocation mechanism no longer works. Our mechanism achieves robustness for such key-exposure: Even if the server secret key is leaked, a revoked user still cannot decrypt any data. In particular, the server secret key and ciphertexts are updated periodically or in emergency situations, e.g. when the server secret key is stolen by hackers.
- **Provable Security.** The proposed scheme does not affect the security of SM9-IBE, which is provably secure against adaptive-ID Chosen Ciphertext Attacks. Furthermore, for the security of revocation, we formally define a concrete security property called *user revocation validity* where a revoked user cannot decrypt any ciphertexts without the help of the server.
- **Performance Evaluation.** We conduct theoretical comparison between the proposed mechanism and related works, showing the advantage and efficiency of SA-IR-RIBE. Moreover, we present comprehensive experimental evaluations by implementing the proposed mechanism. We benchmark our implementations on a high concurrency scenario, where tens of thousands users upload and download data from the server simultaneously. The obtained results indicate that the proposed mechanism is practical for real-world application scenarios.

Limitation. The proposed method deals with the problem of user revocation. It does not directly support key revocation. However, key revocation can be achieved in the cost of changing identities. Specifically, in our system, when a user lost his key, he/she informs the key generation center to revoke his/her key and applies for a new key under a new identity.

Related work

Numerous revocation techniques have been developed in the IBE setting. In this section, we review some existing solutions and identify the problems when they are applied to SM9-IBE.

Generic Time-Concatenated Solution. The first revocable IBE scheme is proposed by Boneh and Franklin (2001) (BF-IBE). At each time period t, KGC issues secret keys $\mathsf{SK}_{\mathsf{ID}_i||\mathsf{t}}$ to all non-revoked users for identities $\mathsf{ID}_i||\mathsf{t}$. The data owners encrypt messages under the new concatenated identities $\mathsf{ID}_i||\mathsf{t}$. Thus, if a user is revoked in time period t, he/she cannot access the messages encrypted in t. However, this mechanism does not scale well, since KGC must re-generate a large number of secret keys and re-distribute them to users at the beginning of each time period. The complexity of KGC is linear in the number of non-revoked users.

Tree-Based Solutions. Boldyreva, Goyal and Kumar (2008) (BGK-RIBE) gave their security notion for revocable IBE (RIBE), and constructed an efficient revocable IBE scheme from the fuzzy IBE scheme (Sahai and Waters 2005) with binary tree structure (Naor et al. 2001) in the selective-ID security model. In the BGK method, each user has a tuple of long term secret keys. KGC publicly broadcasts a small set of key updates in each time period, so that only non-revoked users can construct new decryption keys from their long term secret keys and the key updates. BGK-RIBE significantly reduces the total size of key updates from linear to logarithmic in the number of non-revoked users. Following work by Libert and Vergnaud (2009) proposed an adaptive secure revocable IBE scheme based on the variant of Waters IBE (Waters

2005) and Gentry IBE (Gentry 2006). Recently, Seo and Emura (2013) revisited the security notation of RIBE by presenting the decryption key exposure attack, and proposed a notable scheme based on Libert and Vergnaud (2009). However, above approaches have two limitations: (1) all non-revoked users have to download public key updates from KGC periodically; and (2) the sizes of both users' secret keys and public key updates grow logarithmically in the number of non-revoked users.

Server-Aided Solutions. Several works adopted a third party to achieve revocation in the IBE setting. Boneh et al. (2001); Libert and Quisquater (2003) employed a trusted party called mediator that holds all users' secret keys and helps users to decrypt all ciphertexts. If a user is revoked, the mediator stops helping the user. This model is impractical since users have to fully trust the mediator and they need to communicate with it for each decryption. The work (Li et al. 2013) showed how to outsource workload of the KGC to a semi-trusted server, which they referred as outsourced KGC. In their approach, a secret key is split into two shares held by a user and the server. For a revoked user, the server refuses to collaborate. A disadvantage of (Li et al. 2013) is that the sever needs to maintain all users' secret key shares. Another work (Qin et al. 2015) adapted the scheme of Seo and Emura (2013) to delegate public key update workload from KGC to an untrusted server. Each user keeps one short secret key and does not communicate with the KGC or the server during key updating. Their scheme is provably secure against adaptive-ID chosen ciphertext attacks under the DBDH assumption in the standard model.

Broadcast Encryption. Identity-Based Broadcast Encryption (IBBE) (Delerablée 2007) is a natural generalization of broadcast encryption (BE) in the IBE setting. While adopting BE schemes to support user revocation has been well studied (Naor et al. 2001; Kogan et al. 2006), the IBBE primitive itself does not imply a solution for the user revocation problem. Until very recently, Ge and Wei (2019) formally studied scalable revocation methodology for IBBE schemes. Following the binary tree data structure in Boldyreva et al. (2008), they gave a concrete revocable IBBE scheme, which is semi-adaptively secure under Chosen Plaintext Attacks (CPA) in the standard model. However, the size of the secret keys is linear in the number of maximum size of the recipients in one encryption.

Limitations when Applying These Approaches to SM9-IBE. According to Boyen (2007), known constructions of pairing-based IBE schemes can be classified into 3 families: "Full Domain Hash" IBE (e.g., BF-IBE), "Exponent Inversion" IBE (e.g., SK-IBE (Sakai and Kasahara 2003)) and "Commutative Blinding" IBE (e.g., Waters IBE). A fact must be noted is that all above revocation mechanisms are construed using "Full Domain Hash" IBE or "Commutative Blinding" IBE schemes. While the time-concatenated solution (i.e., BF-IBE) is a generic technique that can be applied to all above 3 families of IBE, other revocation mechanisms (Boldyreva et al. 2008; Libert and Vergnaud 2009; Seo and Emura 2013; Li et al. 2013; Qin et al. 2015; Ge and Wei 2019) relied on the concrete structure of the underlying IBE schemes more or less. However, SM9-IBE is actually an "Exponent Inversion"-like IBE scheme. Therefore, all mentioned revocation mechanisms (except BF-IBE) cannot be trivially adopted for SM9-IBE. As for BF-IBE, it has an unaffordable burden for both KGC and users.

We note that all the above-mentioned revocation mechanisms only achieve indirect user revocation, which means that the revocation needs time to take effect. In such a situation, hackers may have downloaded and decrypted all data during the time period.

Preliminary

Notations. Let $||$ and $\oplus$ denote bitwise operations concatenation and XOR, respectively. We call a function *negl* negligible in λ, if for every positive polynomial $poly(\cdot)$ there exists an N such that for all $\lambda > N$, $negl(\lambda) < 1/poly(\lambda)$. A probabilistic polynomial-time (PPT) algorithm A is an algorithm that on input x, computes $A(x)$ using randomness and its running time is bounded by $poly(\lambda)$. Following primitives are used in both the SM9 and this work. One may refer to ISO/IEC 18033-2 (Shoup 2006) and (Cheng 2017) for detailed definitions.

- BITS(m): Count the bit length of a bit string m.
- EC2OSP(P): Convert an elliptic curve point P to an octet string.
- FE2OSP(W): Convert a field element W to an octet string.
- KDF2(H_v, m, l): Given a hash function H_v with v-bit output, a bit string m and a non-negative integer l, the algorithm derives a l-bit key string.
- H2RF$_i$(H_v, m, n): Given a hash function H_v with v-bit output, a bit string m, a non-negative integer n and a non-negative integer i, the algorithm outputs an integer h_i where $1 \leq h_i \leq n-1$.

Bilinear Pairing. Let $\mathcal{BP}$ be an algorithm that takes as input a security parameter λ and outputs a tuple $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, p, e)$, where $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_T$ are multiplicative cyclic groups of prime order p, and $e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$ is an admissible bilinear map if: 1) Bilinearity: for all $g_1 \in \mathbb{G}_1$, $g_2 \in \mathbb{G}_2$ and $a, b \in \mathbb{Z}_p^*$, we have $e(g_1^a, g_2^b) = e(g_1, g_2)^{ab}$. 2) Non-Degeneracy: $e(g_1, g_2) \neq 1$ whenever $g_1 \neq 1_{\mathbb{G}_1}$ and $g_2 \neq 1_{\mathbb{G}_2}$.

Gap-τ-Bilinear Collision Attack Assumption (Gap-τ-BCAA1$_{i,j}$) (Cheng 2017). For a bilinear pairing $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, p, e)$, $\alpha \in \mathbb{Z}_p^*$, $g_1 \in \mathbb{G}_1$ and $g_2 \in \mathbb{G}_2$, given

$(g_1, g_2, g_i^{\alpha}, h_0, (h_1, g_j^{\frac{\alpha}{h_1+\alpha}}), \ldots, (h_p, g_j^{\frac{\alpha}{h_\tau+\alpha}}))$ for some values $i, j \in \{1, 2\}$ where $h_i \in \mathbb{Z}_p^*$ and different from each other for $0 \leq i \leq \tau$, computing $e(g_1, g_2)^{\frac{\gamma}{h_0+\gamma}}$ is hard.

The SM9 identity-based encryption

The SM9 encryption (SM9-IBE) (Gm/t 2016a; Cheng 2017) is a hybrid encryption scheme that follows the Key/Data Encapsulation Mechanism (KEM/DEM) paradigm (Shoup 2001) in the IBE setting (Bentahar et al. 2008). We first introduce the SM9 key encapsulation mechanism (SM9-KEM), and then present the hybrid encryption scheme. There are four algorithms in SM9-KEM:

$\mathsf{Setup}_{\mathsf{SM9\text{-}KEM}}(\lambda) \rightarrow (\mathsf{MPK}_{\mathsf{SM9}}, \mathsf{MSK}_{\mathsf{SM9}})$. On input a security parameter λ, the algorithm runs as follows:

1. Choose $\mathcal{G} = (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, p, e) \leftarrow \mathcal{BP}(\lambda)$. Pick random generators $g \in \mathbb{G}_1, h \in \mathbb{G}_2$. Pick a random $\gamma \in \mathbb{Z}_p^*$, compute $w = g^{\gamma}$ and $u = e(g, h)^{\gamma}$.
2. Let $F(\mathsf{ID}_i) = \mathrm{H2RF}_1(H_v, \mathsf{ID}_i || hid, p)$, where $hid = 3$ and H_v is a cryptographic hash function with v-bit output.
3. Output $\mathsf{MPK}_{\mathsf{SM9}} = (\mathcal{G}, g, h, u, w, F)$ and $\mathsf{MSK}_{\mathsf{SM9}} = \gamma$.

$\mathsf{Extract}_{\mathsf{SM9\text{-}KEM}}(\mathsf{MPK}_{\mathsf{SM9}}, \mathsf{MSK}_{\mathsf{SM9}}, \mathsf{ID}_i) \rightarrow \mathsf{SK}_{\mathsf{ID}_i}$. On input a master public key $\mathsf{MPK}_{\mathsf{SM9}}$, a master secret key $\mathsf{MSK}_{\mathsf{SM9}}$ and an identity ID_i, the algorithm outputs a secret key $\mathsf{SK}_{\mathsf{ID}_i} = h^{\frac{\gamma}{\gamma+F(\mathsf{ID}_i)}}$.

$\mathsf{Encap}_{\mathsf{SM9\text{-}KEM}}(\mathsf{MPK}_{\mathsf{SM9}}, \mathsf{ID}_i) \rightarrow (\mathsf{CT}_{\mathsf{ID}_i}, K)$. On input a master public key $\mathsf{MPK}_{\mathsf{SM9}}$, an identity ID_i and a message m, the algorithm runs as follows:

1. Select a random $z \in \mathbb{Z}_p^*$. Compute $Q = w \cdot g^{F(\mathsf{ID}_i)} = g^{\gamma+F(\mathsf{ID}_i)}$, $C_1 = Q^z$, $t = u^z$.
2. Derive a session key $K = \mathrm{KDF2}(H_v, \mathrm{EC2OSP}(C_1) || \mathrm{FE2OSP}(t) || \mathsf{ID}_i, k)$ where k is the key length of the DEM part.
3. Output $(\mathsf{CT}_{\mathsf{ID}_i} = C_1, K)$.

$\mathsf{Decap}_{\mathsf{SM9\text{-}KEM}}(\mathsf{MPK}_{\mathsf{SM9}}, \mathsf{SK}_{\mathsf{ID}_i}, \mathsf{CT}_{\mathsf{ID}_i}) \rightarrow K$. On input a master public key $\mathsf{MPK}_{\mathsf{SM9}}$, a secret key $\mathsf{SK}_{\mathsf{ID}_i}$ and a ciphertext $\mathsf{CT}_{\mathsf{ID}_i}$, the algorithm runs as follows:

1. Compute $t = e(\mathsf{CT}_{\mathsf{ID}_i}, \mathsf{SK}_{\mathsf{ID}_i}) = u^z$.
2. Derive the encapsulated key $K = \mathrm{KDF2}(H_v, \mathrm{EC2OSP}(\mathsf{CT}_{\mathsf{ID}_i}) || \mathrm{FE2OSP}(t) || \mathsf{ID}_i, k)$.
3. Output the key K.

Let $\mathcal{E}_{\mathsf{sym}} = (\mathsf{Enc}_{\mathsf{sym}}, \mathsf{Dec}_{\mathsf{sym}})$ denote a symmetric encryption scheme that consists of an encryption algorithm $\mathsf{Enc}_{\mathsf{sym}}$ and a decryption algorithm $\mathsf{Dec}_{\mathsf{sym}}$, where: (1) $\mathsf{Enc}_{\mathsf{sym}}$ takes input a key and a message, and outputs a ciphertext; and (2) $\mathsf{Dec}_{\mathsf{sym}}$ takes input a key and a ciphertext, and outputs a message. The full SM9-IBE encryption scheme is constructed as follows:

$\mathsf{Setup}_{\mathsf{SM9}}(\lambda) \rightarrow (\mathsf{MPK}_{\mathsf{SM9}}, \mathsf{MSK}_{\mathsf{SM9}})$. It is the same as the algorithm $\mathsf{Setup}_{\mathsf{SM9\text{-}KEM}}$ in SM9-KEM.

$\mathsf{Extract}_{\mathsf{SM9}}(\mathsf{MPK}_{\mathsf{SM9}}, \mathsf{MSK}_{\mathsf{SM9}}, \mathsf{ID}_i) \rightarrow \mathsf{SK}_{\mathsf{ID}_i}$. It is the same as the algorithm $\mathsf{Extract}_{\mathsf{SM9\text{-}KEM}}$ in SM9-KEM.

$\mathsf{Enc}_{\mathsf{SM9}}(\mathsf{MPK}_{\mathsf{SM9}}, \mathsf{ID}_i, m) \rightarrow \mathsf{CT}_{\mathsf{ID}_i}$. On input a master public key $\mathsf{MPK}_{\mathsf{SM9}}$, an identity ID_i and a message m, the algorithm runs $\mathsf{Encap}_{\mathsf{SM9\text{-}KEM}}(\mathsf{MPK}_{\mathsf{SM9}}, \mathsf{ID}_i)$ to obtain (C_1, K). Further, the message is encrypted using the symmetric encryption algorithm: $C_2 = \mathsf{Enc}_{\mathsf{sym}}(K, m)$. The algorithm outputs $\mathsf{CT}_{\mathsf{ID}_i} = (C_1, C_2)$.

$\mathsf{Dec}_{\mathsf{SM9}}(\mathsf{MPK}_{\mathsf{SM9}}, \mathsf{SK}_{\mathsf{ID}_i}, \mathsf{CT}_{\mathsf{ID}_i}) \rightarrow m$. On input a master public key $\mathsf{MPK}_{\mathsf{SM9}}$, a secret key $\mathsf{SK}_{\mathsf{ID}_i}$ and a ciphertext $\mathsf{CT}_{\mathsf{ID}_i} = (C_1, C_2)$, the algorithm runs $\mathsf{Decap}_{\mathsf{SM9\text{-}KEM}}(\mathsf{MPK}_{\mathsf{SM9}}, \mathsf{SK}_{\mathsf{ID}_i}, C_1)$ to obtain the encapsulated K. The DEM part is decrypted using the symmetric decryption algorithm: $m = \mathsf{Dec}_{\mathsf{sym}}(K, C_2)$. The algorithm outputs the message m.

Chosen Ciphertext Security of SM9-IBE. In the identity-based hybrid encryption setting, Bentahar et al. (2008) proved that a hybrid IBE scheme is secure against adaptive-ID Chosen Ciphertext Attacks (ID-IND-CCA), if the underlying KEM part is ID-IND-CCA secure and the DEM part is a one-time symmetric encryption scheme that can resistant Find-Guess (FG) Chosen Ciphertext Attacks (FG-CCA) (cf. Theorem 1 in Bentahar et al. (2008)). Following this framework, (Cheng 2017) first proved that SM9-KEM is ID-IND-CCA secure with the Gap-p-BCAA1$_{1,2}$ assumption in the random oracle model, and then stated that SM9-IBE is IND-ID-CCA secure giving an FG-CCA secure DEM.

The model of SA-IR-RIBE

In this section, we present the system model and the security definitions.

System model

In the proposed mechanism, there are four parties involved: a key generation center (KGC), a server, data owners and data receivers. The compact system model is illustrated in Fig. 1. All acronyms used in this paper are listed in Table 1.

KGC is responsible for generating and distributing system parameters to other entities, including the master public key, the master secret key, the server secret key and users' secret keys. Besides, it maintains a revocation list for realizing user revocation. The server provides computing and storage resources for users. It gets ciphertexts from data owners and pushes the ciphertexts to data receivers. Most importantly, the server accomplish the user revocation functionality according to the revocation list that provided by KGC. We note that an existing server can be adopted to play the role of the server in our system, e.g., a SMTP server in an email system. Data owners encrypt messages under data receivers' identities and

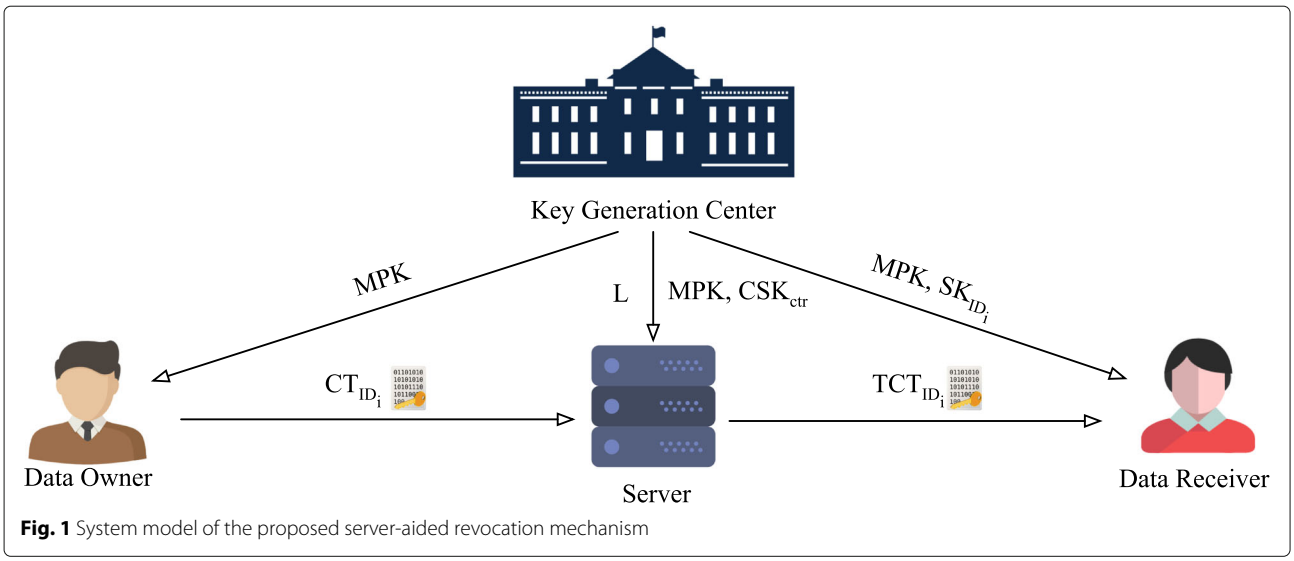

Fig. 1 System model of the proposed server-aided revocation mechanism

uploads ciphertexts to the server. Data receivers obtain ciphertexts from the server and decrypt them with their secret keys.

The Server-Aided Immediate and Robust Revocable Identity-Based Encryption (SA-IR-RIBE) scheme consists of 9 algorithms:

- $\mathsf{Setup}(\lambda) \rightarrow (\mathsf{MPK}, \mathsf{MSK}, \mathsf{SSK}_{\mathsf{ctr}}, \mathbb{L})$. On input a security parameter λ, the algorithm (run by KGC) outputs a master public key MPK, a master secret key MSK, an initial server secret key $\mathsf{SSK}_{\mathsf{ctr}}$ and a revocation list $\mathbb{L}$.
- $\mathsf{Extract}(\mathsf{MPK}, \mathsf{MSK}, \mathsf{ID}_i) \rightarrow \mathsf{SK}_{\mathsf{ID}_i}$. On input a master public key MPK, a master secret key MSK, and an identity ID_i, the algorithm (run by KGC) outputs a secret key $\mathsf{SK}_{\mathsf{ID}_i}$.
- $\mathsf{Enc}(\mathsf{MPK}, \mathsf{ID}_i, m) \rightarrow \mathsf{CT}_{\mathsf{ID}_i}$. On input a master public key MPK, an identity ID_i and a message m, the algorithm (run by a data owner) outputs a ciphertext $\mathsf{CT}_{\mathsf{ID}_i}$.
- $\mathsf{CTInit}(\mathsf{MPK}, \mathsf{CT}_{\mathsf{ID}_i}, \mathsf{SSK}_{\mathsf{ctr}}) \rightarrow \mathsf{CT}_{\mathsf{ID}_i,\mathsf{ctr}}$. On input a master public key MPK, a ciphertext $\mathsf{CT}_{\mathsf{ID}_i}$ and a server secret key $\mathsf{SSK}_{\mathsf{ctr}}$, the algorithm (run by the server) outputs an updated $\mathsf{CT}_{\mathsf{ID}_i,\mathsf{ctr}}$.
- $\mathsf{SSKUpdate}(\mathsf{MPK}, \mathsf{SSK}_{\mathsf{ctr}}) \rightarrow \mathsf{SSK}_{\mathsf{ctr}'}$. On input a master public key MPK and a server secret key $\mathsf{SSK}_{\mathsf{ctr}}$, the algorithm (run by the server) outputs an updated server secret key $\mathsf{SSK}_{\mathsf{ctr}'}$.
- $\mathsf{CTUpdate}(\mathsf{MPK}, \mathsf{CT}_{\mathsf{ID}_i,\mathsf{ctr}}, \mathsf{SSK}_{\mathsf{ctr}}, \mathsf{SSK}_{\mathsf{ctr}'}) \rightarrow \mathsf{CT}_{\mathsf{ID}_i,\mathsf{ctr}'}$. On input a master public key MPK, a ciphertext $\mathsf{CT}_{\mathsf{ID}_i,\mathsf{ctr}}$, and two server secret keys $\mathsf{SSK}_{\mathsf{ctr}}, \mathsf{SSK}_{\mathsf{ctr}'}$, the algorithm (run by the server) outputs an updated ciphertext $\mathsf{CT}_{\mathsf{ID}_i,\mathsf{ctr}'}$.
- $\mathsf{Transform}(\mathsf{MPK}, \mathsf{CT}_{\mathsf{ID}_i,\mathsf{ctr}}, \mathsf{SSK}_{\mathsf{ctr}}, \mathbb{L}) \rightarrow \mathsf{TCT}_{\mathsf{ID}_i}$. On input a master public key MPK, a ciphertext $\mathsf{CT}_{\mathsf{ID}_i,\mathsf{ctr}}$, a server key $\mathsf{SSK}_{\mathsf{ctr}}$ and a list $\mathbb{L}$, the algorithm (run by the server) outputs a transformed ciphertext $\mathsf{TCT}_{\mathsf{ID}_i}$.
- $\mathsf{Dec}(\mathsf{MPK}, \mathsf{TCT}_{\mathsf{ID}_i}, \mathsf{SK}_{\mathsf{ID}_i}) \rightarrow m$. On input a master public key MPK, a transformed ciphertext $\mathsf{TCT}_{\mathsf{ID}_i}$, and a secret key $\mathsf{SK}_{\mathsf{ID}_i}$, the algorithm (run by a data receiver) outputs a message m.
- $\mathsf{Revoke}(\mathsf{ID}_i, \mathbb{L}) \rightarrow \mathbb{L}'$. On input an identity ID_i and a revocation list $\mathbb{L}$, the algorithm (run by KGC) outputs an updated revocation list $\mathbb{L}'$. The new list $\mathbb{L}'$ is further distributed to the server.

The players in the system may conduct the following interactions (cf. Fig. 1):

Table 1 Acronyms used in this paper

Acronym	Description	Acronym	Description
MPK	master public key	MSK	master secret key
ID_i	the i-th user's identity	$\mathsf{SK}_{\mathsf{ID}_i}$	the i-th user's secret key
ctr	the time counter	$\mathsf{SSK}_{\mathsf{ctr}}$	the server secret key related to ctr
$\mathbb{L}$	the user revocation list	m	message to be encrypted
K	the derived symmetric key	$\mathsf{CT}_{\mathsf{ID}_i}$	ciphertext that sent to the i-th user
$\mathsf{CT}_{\mathsf{ID}_i,\mathsf{ctr}}$	ciphertext that sent to i-th user related to ctr	$\mathsf{TCT}_{\mathsf{ID}_i}$	transformed ciphertext that sent to i-th user

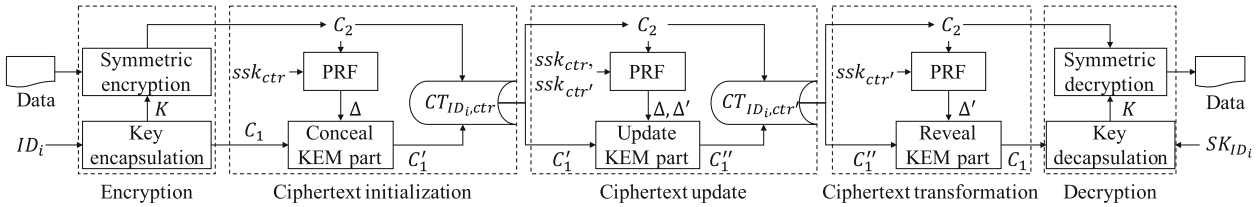

Fig. 2 Overview of the proposed server-aided revocation mechanism

System initialization: PKG runs the algorithms Setup and Extract to generate system parameters and all users' secret keys. It then distributes MPK to all other parties, $\mathsf{SSK}_{\mathsf{ctr}}$ and $\mathbb{L}$ to the server and $\mathsf{SK}_{\mathsf{ID}_i}$ to users. Initially, the user revocation list $\mathbb{L}$ is empty.

Message encryption: A data owner runs the algorithm Enc to encrypt a message under an identity ID_i, and obtains a ciphertext $\mathsf{CT}_{\mathsf{ID}_i}$. The ciphertext $\mathsf{CT}_{\mathsf{ID}_i}$ is uploaded to the server and then concealed by the server with the algorithm CTInit using the server secret key $\mathsf{SSK}_{\mathsf{ctr}}$.

Server key evolution: To resist exposure of the server secret key, the server updates the server secret key $\mathsf{SSK}_{\mathsf{ctr}}$ and all stored ciphertexts periodically or when the server secret key leaks. When the server secret key $\mathsf{SSK}_{\mathsf{ctr}}$ is demands to be updated, a new server secret key $\mathsf{SSK}_{\mathsf{ctr}'}$ is randomly sampled by calling the algorithm SSKUpdate. Meanwhile, all stored ciphertext are updated with the algorithm CTUpdate using the two server secret keys $\mathsf{SSK}_{\mathsf{ctr}}, \mathsf{SSK}_{\mathsf{ctr}'}$, i.e., $\mathsf{CT}_{\mathsf{ID}_i,\mathsf{ctr}}$ is updated to $\mathsf{CT}_{\mathsf{ID}_i,\mathsf{ctr}'}$. Once all ciphertexts have been updated, $\mathsf{SSK}_{\mathsf{ctr}'}$ becomes the current server secret key. Meanwhile, all ciphertexts in time ctr and $\mathsf{SSK}_{\mathsf{ctr}}$ are erased from the server's storage.

Message decryption: Before pushing a stored ciphertext to a data receiver, the server runs Transform to obtain a transformed ciphertext $\mathsf{TCT}_{\mathsf{ID}_i}$. On receiving $\mathsf{TCT}_{\mathsf{ID}_i}$, the receiver decrypts it with the algorithm Dec using his/her secret key. Note that the algorithm Transform takes the revocation list $\mathbb{L}$ as its input, thus the ciphertext $\mathsf{TCT}_{\mathsf{ID}_i}$ is computed in a manner that only non-revoked receivers can decrypt the ciphertext successfully.

User revocation: When a user is required to be revoked, KGC updates the list $\mathbb{L}$ by running the algorithm Revoke and sends the updated list $\mathbb{L}$ to the server.

Security definition

Adversarial Model. KGC generates user secret keys, and are assumed to be honest. Data owners are trusted and we do not consider the data they own as correct or incorrect. The server is honest-but-curious (Li et al. 2012), meaning that it will honestly follow the protocol but try to learn as much information as possible. This assumption is realistic, since the server will be a service provider that cares its reputation and thus restricted by user contrast. While most of data receivers are trusted, some of them are corrupt and share their secret keys in the collusion. This collusion may gives the adversary more power.

Chosen Ciphertext Security. We give the semantic security against adaptively chosen-ID and Chosen Ciphertext Attacks for server-aided revocable IBE scheme (in short, SA-IND-ID-CCA).

The adversary is able to obtain a set of corrupted users' secret keys $\{\mathsf{SK}_{\mathsf{ID}_1}, \mathsf{SK}_{\mathsf{ID}_2}, \ldots\}$ (except the target), and all the server secret keys, i.e., $\mathsf{SSK}_1, \mathsf{SSK}_2, \ldots, \mathsf{SSK}_{\mathsf{ctr}}$. The security is defined with the following game:

Setup: The challenger $\mathcal{C}$ takes a security parameter λ and runs Setup algorithm. It gives $\mathsf{MPK}, \mathsf{SSK}_{\mathsf{ctr}}$ to the adversary $\mathcal{A}$ and keeps MSK to itself. The current time counter ctr is set to 1. $\mathcal{C}$ periodically updates the counter to ctr′ and sends the new server secret key $\mathsf{SSK}_{\mathsf{ctr}'}$ to $\mathcal{A}$.

Phase 1: $\mathcal{A}$ adaptively issues queries to following three oracles:

- $\mathcal{O}_{\mathsf{Extract}}(\mathsf{ID}_i)$: $\mathcal{C}$ sends a secret key $\mathsf{SK}_{\mathsf{ID}_i}$ to $\mathcal{A}$, where $\mathsf{SK}_{\mathsf{ID}_i}$ is generated with the algorithm Extract.
- $\mathcal{O}_{\mathsf{Dec}}(\mathsf{ID}_i, \mathsf{CT}_{\mathsf{ID}_i,\mathsf{ctr}})$: $\mathcal{C}$ obtains $\mathsf{SK}_{\mathsf{ID}_i}$ by running algorithm Extract. It then decrypts $\mathsf{CT}_{\mathsf{ID}_i,\mathsf{ctr}}$ by running algorithms Transform and Dec to obtain the plaintext m. $\mathcal{C}$ outputs m to $\mathcal{A}$.
- $\mathcal{O}_{\mathsf{SSK}}(\mathsf{ctr}_i)$: If the server secret key $\mathsf{SSK}_{\mathsf{ctr}_i}$ has not been generated, $\mathcal{C}$ randomly samples $\mathsf{SSK}_{\mathsf{ctr}_i}$ and stores $\mathsf{SSK}_{\mathsf{ctr}_i}$. Otherwise, $\mathcal{C}$ recalls $\mathsf{SSK}_{\mathsf{ctr}_i}$ from its storage. $\mathcal{C}$ outputs $\mathsf{SSK}_{\mathsf{ctr}_i}$ to $\mathcal{A}$.

Challenge: Once $\mathcal{A}$ decides that **Phase 1** is over, it outputs a challenge identity ID^*, a challenge time counter ctr^* and two equal length messages m_0, m_1. Note that ID^* did not appear in the previous queries to $\mathcal{O}_{\mathsf{Extract}}$. If $\mathsf{SSK}_{\mathsf{ctr}^*}$ has not been generated, $\mathcal{B}$ randomly samples $\mathsf{SSK}_{\mathsf{ctr}^*}$. The challenger $\mathcal{C}$ chooses a fair coin $\mu \in \{0, 1\}$ and runs algorithm $\mathsf{Enc}(\mathsf{MPK}, \mathsf{ID}^*, m_\mu)$ to obtain $\mathsf{CT}^*_{\mathsf{ID}^*}$. Further, it runs $\mathsf{CTInit}(\mathsf{MPK}, \mathsf{CT}^*_{\mathsf{ID}^*}, \mathsf{SSK}_{\mathsf{ctr}^*})$ to obtain $\mathsf{CT}^*_{\mathsf{ID}^*,\mathsf{ctr}^*}$. The challenger returns $\mathsf{CT}^*_{\mathsf{ID}^*,\mathsf{ctr}^*}$ as the challenge ciphertext to the adversary $\mathcal{A}$.

Phase 2: $\mathcal{A}$ adaptively issues more queries to the three oracles:

- $\mathcal{O}_{\mathsf{Extract}}(\mathsf{ID}_i)$ where $\mathsf{ID}_i \neq \mathsf{ID}^*$: The challenger responds as in **Phase 1**.
- $\mathcal{O}_{\mathsf{Dec}}(\mathsf{ID}_i, \mathsf{CT}_{\mathsf{ID}_i,\mathsf{ctr}})$ where

$(\mathsf{ID}_i, \mathsf{CT}_{\mathsf{ID}_i,\mathsf{ctr}}) \neq (\mathsf{ID}^*, \mathsf{CT}^*_{\mathsf{ID}^*,\mathsf{ctr}^*})$: The challenger responds as in **Phase 1**.

- $\mathcal{O}_{\mathsf{SSK}}(\mathsf{ctr}_i)$: The challenger responds as in **Phase 1**.

Guess: Finally, the adversary $\mathcal{A}$ outputs a guess $\mu' \in \{0,1\}$ and wins the game if $\mu = \mu'$.

Definition 1 (SA-IND-ID-CCA Security) *We say that a SA-IR-RIBE scheme is* SA-IND-ID-CCA *secure if for any PPT adversary $\mathcal{A}$ the function $Adv_{\mathcal{A}}$ is negligible:*

$$Adv_{\mathcal{A}} = |Pr[\mu = \mu'] - \frac{1}{2}| \leq negl(\lambda).$$

User Revocation Validity. An important security property of a SA-IR-RIBE scheme is user revocation validity. Namely, if a user is revoked in time ctr^*, he/she cannot decrypt all ciphertexts that encrypted to his/her identity. In this situation, the adversary is the revoked user with the challenge identity ID^*, and holds the corresponding secret key $\mathsf{SK}_{\mathsf{ID}^*}$. Meanwhile, the adversary is allowed to obtain a set of corrupted users' secret keys, i.e., $\{\mathsf{SK}_{\mathsf{ID}_1}, \mathsf{SK}_{\mathsf{ID}_2}, \ldots\}$. Besides, in time ctr^*, the server may discard all previous server secret keys, i.e., $\mathsf{SSK}_1, \mathsf{SSK}_2, \ldots, \mathsf{SSK}_{\mathsf{ctr}^*-1}$ and only protect its current secret key $\mathsf{SSK}_{\mathsf{ctr}^*}$. Thus, the adversary may obtain all previous server secret keys except the one in the challenge time counter, i.e., $\mathsf{SSK}_{\mathsf{ctr}^*}$. We remark that this assumption gives the adversary maximum power to attack the system. Semantic security against adaptive-ID Chosen Plaintext Attacks (CPA) for User Revocation Validity is defined with the following game:

Setup: The challenger $\mathcal{C}$ takes a security parameter λ and runs Setup algorithm. It gives MPK to the adversary $\mathcal{A}$ and keeps $\mathsf{SSK}_{\mathsf{ctr}}$ and MSK to itself. The current time counter ctr is set to 1. $\mathcal{C}$ periodically updates the counter to ctr' and keeps $\mathsf{SSK}_{\mathsf{ctr}'}$ to itself.

Phase 1: $\mathcal{A}$ adaptively issues queries to following two oracles:

- $\mathcal{O}_{\mathsf{Extract}}(\mathsf{ID}_i)$: $\mathcal{C}$ sends a secret key $\mathsf{SK}_{\mathsf{ID}_i}$ to $\mathcal{A}$, where $\mathsf{SK}_{\mathsf{ID}_i}$ is generated with algorithm Extract.
- $\mathcal{O}_{\mathsf{SSK}}(\mathsf{ctr}_i)$: If the server secret key $\mathsf{SSK}_{\mathsf{ctr}_i}$ has not been generated, $\mathcal{C}$ randomly samples $\mathsf{SSK}_{\mathsf{ctr}_i}$ and stores $\mathsf{SSK}_{\mathsf{ctr}_i}$. Otherwise, $\mathcal{C}$ recalls $\mathsf{SSK}_{\mathsf{ctr}_i}$ from its storage. $\mathcal{C}$ outputs $\mathsf{SSK}_{\mathsf{ctr}_i}$ to $\mathcal{A}$.

Challenge: Once $\mathcal{A}$ decides that **Phase 1** is over, it outputs a challenge identity ID^*, a challenge time counter ctr^* and two equal length messages m_0, m_1. Note that two constraints must be satisfied: (1) ID^* cannot appear in previous queries to $\mathcal{O}_{\mathsf{Extract}}$; and (2) ctr^* cannot appear in previous queries to $\mathcal{O}_{\mathsf{SSK}}$. If the server secret key $\mathsf{SSK}_{\mathsf{ctr}^*}$ has not been generated, $\mathcal{C}$ randomly samples $\mathsf{SSK}_{\mathsf{ctr}^*}$ and keeps the key to itself. $\mathcal{C}$ chooses a fair coin $\mu \in \{0,1\}$, runs algorithm $\mathsf{Enc}(\mathsf{MPK}, \mathsf{ID}^*, m_\mu)$ to obtain $\mathsf{CT}^*_{\mathsf{ID}^*}$. Further, it runs $\mathsf{CTInit}(\mathsf{MPK}, \mathsf{CT}^*_{\mathsf{ID}^*}, \mathsf{SSK}_{\mathsf{ctr}^*})$ to obtain $\mathsf{CT}^*_{\mathsf{ID}^*,\mathsf{ctr}^*}$. The challenger returns $\mathsf{CT}^*_{\mathsf{ID}^*,\mathsf{ctr}^*}$ as the challenge ciphertext to the adversary $\mathcal{A}$.

Phase 2: $\mathcal{A}$ adaptively issues more queries to the three oracles:

- $\mathcal{O}_{\mathsf{Extract}}(\mathsf{ID}_i)$: The challenger responds as in **Phase 1**.
- $\mathcal{O}_{\mathsf{SSK}}(\mathsf{ctr}_i)$ where $\mathsf{ctr}_i \neq \mathsf{ctr}^*$: The challenger responds as in **Phase 1**.

Guess: Finally, the adversary $\mathcal{A}$ outputs a guess $\mu' \in \{0,1\}$ and wins the game if $\mu = \mu'$.

Definition 2 (User Revocation Validity) *We say that a SA-IR-RIBE scheme is adaptive-ID CPA-secure for User Revocation Validity if for any PPT adversary $\mathcal{A}$ the function $Adv_{\mathcal{A}}$ is negligible:*

$$Adv_{\mathcal{A}} = |Pr[\mu = \mu'] - \frac{1}{2}| \leq negl(\lambda).$$

Server-aided immediate and robust revocation for SM9-IBE

In this section, we present the proposed revocation mechanism, which is illustrated in Figs. 2 and 3.

A data owner encrypts data under an identity ID_i, and obtains a ciphertext $\mathsf{CT}_{\mathsf{ID}_i} = (C_1, C_2)$. The encryption algorithm works in the KEM/DEM setting, where C_1 is the KEM part and C_2 is the DEM part. On receiving the ciphertext $\mathsf{CT}_{\mathsf{ID}_i}$, the server takes the following actions: (1) derives a mask Δ from $\mathsf{SSK}_{\mathsf{ctr}}$ and C_2 using a pseudo-random function PRF, (2) masks C_1 to C_1' using Δ, and (3) stores (C_1', C_2) as $\mathsf{CT}_{\mathsf{ID}_i,\mathsf{ctr}}$. Note that the server does not store the original ciphertext $\mathsf{CT}_{\mathsf{ID}_i}$. When updated to a new server secret key $\mathsf{SSK}_{\mathsf{ctr}'}$, all stored ciphertext are updated using the two server secret keys $\mathsf{SSK}_{\mathsf{ctr}}, \mathsf{SSK}_{\mathsf{ctr}'}$. On pushing ciphertext to a data receiver, if the receiver is not revoked, the server transforms ciphertext $\mathsf{CT}_{\mathsf{ID}_i,\mathsf{ctr}'}$ to $\mathsf{TCT}_{\mathsf{ID}_i}$ using $\mathsf{SSK}_{\mathsf{ctr}'}$ and outputs $\mathsf{TCT}_{\mathsf{ID}_i}$ thus the receiver can decrypt $\mathsf{TCT}_{\mathsf{ID}_i}$ to get the plaintext; otherwise, the server returns the concealed ciphertext $\mathsf{CT}_{\mathsf{ID}_i,\mathsf{ctr}'}$.

The identity space, message space, and ciphertext space are $\{0,1\}^*$, $\{0,1\}^*$ and $(\mathbb{G}_T \times \{0,1\}^*)$, respectively. The proposed SA-IR-RIBE scheme consists of the following algorithms:

$\mathsf{Setup}(\lambda) \rightarrow (\mathsf{MPK}, \mathsf{MSK}, \mathsf{SSK}_{\mathsf{ctr}}, \mathbb{L})$. On input a security parameter λ, the algorithm does the following:

1 Choose $\mathcal{G} = (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, p, e) \leftarrow \mathcal{BP}(\lambda)$. Choose random generators $g \in \mathbb{G}_1, h \in \mathbb{G}_2$ and random elements $\gamma \in \mathbb{Z}_p^*$. Set $u = e(g,h)^\gamma$ and $w = g^\gamma$.
2 Let $F(\mathsf{ID}_i) = \mathrm{H2RF}_1(H_v, \mathsf{ID}_i || hid, p)$, where $hid = 3$, and H_v is a cryptographic hash function of v-bit output. Choose a collision resistant hash function $H : \{0,1\}^* \rightarrow \{0,1\}^{l_2}$. Choose a pseudo-random function PRF: $\{0,1\}^{l_1} \times \{0,1\}^{l_2} \rightarrow \mathbb{Z}_p^*$.

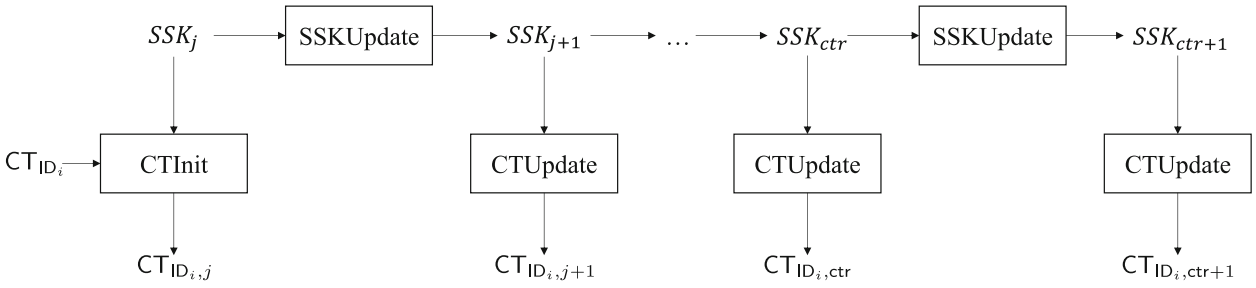

Fig. 3 Overview of the server key evolution and ciphertext update

3 Choose an FG-CCA secure one-time symmetric encryption scheme $\mathcal{E}_{\mathsf{sym}} = (\mathsf{Enc}_{\mathsf{Sym}}, \mathsf{Dec}_{\mathsf{Sym}})$. The key length of $\mathcal{E}_{\mathsf{sym}}$ is k.
4 Choose a random $\mathsf{SSK}_1 \in \{0,1\}^{l_1}$ and set the counter $\mathsf{ctr} = 1$.
5 Return $\mathsf{MPK} = (\mathcal{G}, g, h, u, w, F, H, \mathrm{PRF}, \mathcal{E}_{\mathsf{Sym}})$, $\mathsf{MSK} = \gamma$, $\mathsf{SSK}_{\mathsf{ctr}} = (\mathsf{ssk}_1, \mathsf{ctr})$ and $\mathbb{L} = \emptyset$.

$\mathsf{Extract}(\mathsf{MPK}, \mathsf{MSK}, \mathsf{ID}_i) \rightarrow \mathsf{SK}_{\mathsf{ID}_i}$. On input a master public key MPK, a master secret key MSK, and an identity ID_i, the algorithm outputs a secret key $\mathsf{SK}_{\mathsf{ID}_i} = h^{\frac{\gamma}{\gamma + F(\mathsf{ID}_i)}}$.

$\mathsf{Enc}(\mathsf{MPK}, \mathsf{ID}_i, m) \rightarrow \mathsf{CT}_{\mathsf{ID}_i}$. On input a master public key MPK, an identity ID_i and a message m, the algorithm does the following:

1 Choose a random $z \in \mathbb{Z}_p^*$. Compute $Q = w \cdot g^{F(\mathsf{ID}_i)} = g^{\gamma + F(\mathsf{ID}_i)}, C_1 = Q^z, t = u^z$.
2 Derive a symmetric key $K = \mathrm{KDF2}(H_v, \mathrm{EC2OSP}(C_1) || \mathrm{FE2OSP}(t) || \mathsf{ID}_i, k)$.
3 Compute $C_2 = \mathsf{Enc}_{\mathsf{sym}}(K, m)$.

The algorithm outputs $\mathsf{CT}_{\mathsf{ID}_i} = (C_1, C_2)$.

$\mathsf{CTInit}(\mathsf{MPK}, \mathsf{CT}_{\mathsf{ID}_i}, \mathsf{SSK}_{\mathsf{ctr}}) \rightarrow \mathsf{CT}_{\mathsf{ID}_i,\mathsf{ctr}}$. On input a master public key MPK, a ciphertext $\mathsf{CT}_{\mathsf{ID}_i} = (C_1, C_2)$ and a server secret key $\mathsf{SSK}_{\mathsf{ctr}} = (\mathsf{ssk}_{\mathsf{ctr}}, \mathsf{ctr})$, the algorithm outputs $\mathsf{CT}_{\mathsf{ID}_i,\mathsf{ctr}} = (C_1', C_2)$, where:

$$\Delta = \mathrm{PRF}(\mathsf{ssk}_{\mathsf{ctr}}, H(C_2)), \quad C_1' = C_1^{\Delta}.$$

$\mathsf{SSKUpdate}(\mathsf{MPK}, \mathsf{SSK}_{\mathsf{ctr}}) \rightarrow \mathsf{SSK}_{\mathsf{ctr}'}$. On input a master public key MPK and a server secret key $\mathsf{SSK}_{\mathsf{ctr}} = (\mathsf{ssk}_{\mathsf{ctr}}, \mathsf{ctr})$, the algorithm sets a new counter $\mathsf{ctr}' = \mathsf{ctr} + 1$ and chooses a random $\mathsf{ssk}_{\mathsf{ctr}'} \in \{0,1\}^{l_1}$. The algorithm outputs an updated secret key $\mathsf{SSK}_{\mathsf{ctr}'} = (\mathsf{ssk}_{\mathsf{ctr}'}, \mathsf{ctr}')$.

$\mathsf{CTUpdate}(\mathsf{MPK}, \mathsf{CT}_{\mathsf{ID}_i,\mathsf{ctr}}, \mathsf{SSK}_{\mathsf{ctr}}, \mathsf{SSK}_{\mathsf{ctr}'}) \rightarrow \mathsf{CT}_{\mathsf{ID}_i,\mathsf{ctr}'}$. On input a master public key MPK, a ciphertext $\mathsf{CT}_{\mathsf{ID}_i,\mathsf{ctr}} = (C_1, C_2)$, and two server secret keys $\mathsf{SSK}_{\mathsf{ctr}} = (\mathsf{ssk}_{\mathsf{ctr}}, \mathsf{ctr}), \mathsf{SSK}_{\mathsf{ctr}'} = (\mathsf{ssk}_{\mathsf{ctr}'}, \mathsf{ctr}')$, the algorithm computes:

$$\Delta = \mathrm{PRF}(\mathsf{ssk}_{\mathsf{ctr}}, H(C_2)), \quad \Delta' = \mathrm{PRF}(\mathsf{ssk}_{\mathsf{ctr}'}, H(C_2)),$$
$$C_1' = C_1^{\Delta'/\Delta}.$$

The algorithm outputs an updated ciphertext $\mathsf{CT}_{\mathsf{ID}_i,\mathsf{ctr}'} = (C_1', C_2)$.

$\mathsf{Transform}(\mathsf{MPK}, \mathsf{CT}_{\mathsf{ID}_i,\mathsf{ctr}}, \mathsf{SSK}_{\mathsf{ctr}}, \mathbb{L}) \rightarrow \mathsf{TCT}_{\mathsf{ID}_i}$. On input a master public key MPK, a ciphertext $\mathsf{CT}_{\mathsf{ID}_i,\mathsf{ctr}} = (C_1, C_2)$, a server secret key $\mathsf{SSK}_{\mathsf{ctr}} = (\mathsf{ssk}_{\mathsf{ctr}}, \mathsf{ctr})$ and a revocation list $\mathbb{L}$, if $\mathsf{ID}_i \in \mathbb{L}$, the algorithm returns $\mathsf{CT}_{\mathsf{ID}_i,\mathsf{ctr}}$ as $\mathsf{TCT}_{\mathsf{ID}_i}$; otherwise it computes:

$$\Delta = \mathrm{PRF}(\mathsf{ssk}_{\mathsf{ctr}}, H(C_2)), \quad C_1' = C_1^{1/\Delta} = g^{z(\gamma + F(\mathsf{ID}_i))}.$$

The algorithm outputs a transformed ciphertext $\mathsf{TCT}_{\mathsf{ID}_i} = (C_1', C_2)$.

$\mathsf{Dec}(\mathsf{MPK}, \mathsf{TCT}_{\mathsf{ID}_i}, \mathsf{SK}_{\mathsf{ID}_i}) \rightarrow m$. On input a master public key MPK, a ciphertext $\mathsf{TCT}_{\mathsf{ID}_i} = (C_1, C_2)$, and a secret key $\mathsf{SK}_{\mathsf{ID}_i}$, the algorithm computes $t = e(C_1, \mathsf{SK}_{\mathsf{ID}_i}) = e(g,h)^{\gamma z} = u^z$. The encapsulated key is $K = \mathrm{KDF2}(H_v, \mathrm{EC2OSP}(C_1) || \mathrm{FE2OSP}(t) || \mathsf{ID}_i, k)$. The ciphertext C_2 is decrypted to m with the symmetric decryption algorithm: $m = \mathsf{Dec}_{\mathsf{sym}}(K, C_2)$.

$\mathsf{Revoke}(\mathsf{ID}_i, \mathbb{L}) \rightarrow \mathbb{L}'$. On input an identity ID_i and a revocation list $\mathbb{L}$, the algorithm outputs an updated revocation list $\mathbb{L}' = \mathbb{L} \cup \{\mathsf{ID}_i\}$.

Security proof

Theorem 1 *The proposed server-aided Identity-Based Encryption scheme achieves* SA-IND-ID-CCA *security with respect to Definition 1, if the SM9-IBE scheme is* IND-ID-CCA *secure.*

Proof Suppose the adversary $\mathcal{A}$ can break the proposed SA-IR-RIBE scheme with a non-negligible advantage, we build a simulator $\mathcal{B}$ to break the security of the SM9-IBE scheme with a non-negligible advantage. $\mathcal{B}$ interacts with $\mathcal{A}$ as the challenger. Let $\mathcal{C}$ be the challenger in the IND-ID-CCA secure game of the SM9-IBE scheme. It provides two oracles to $\mathcal{B}$:

- $\mathcal{O}^{\mathsf{SM9}}_{\mathsf{Extract}}(\mathsf{ID}_i)$: $\mathcal{C}$ runs $\mathsf{Extract}_{\mathsf{SM9}}(\mathsf{ID}_i)$ and returns the secret key $\mathsf{SK}_{\mathsf{ID}_i}$.
- $\mathcal{O}^{\mathsf{SM9}}_{\mathsf{Dec}}(\mathsf{ID}_i, \mathsf{CT}_{\mathsf{ID}_i})$: $\mathcal{C}$ runs $\mathsf{Extract}_{\mathsf{SM9}}(\mathsf{ID}_i)$ to obtain $\mathsf{SK}_{\mathsf{ID}_i}$. It then runs $\mathsf{Dec}_{\mathsf{SM9}}(\mathsf{MPK}, \mathsf{SK}_{\mathsf{ID}_i}, \mathsf{CT}_{\mathsf{ID}_i})$ to decrypt $\mathsf{CT}_{\mathsf{ID}_i}$. The resulting plaintext is sent to $\mathcal{B}$.

The simulator $\mathcal{B}$ works by interacting with the challenger $\mathcal{C}$ and the adversary $\mathcal{A}$ as follows:

Setup. $\mathcal{C}$ sends $\mathsf{MPK}_{\mathsf{SM9}} = (\mathcal{G}, g, h, u, w, F, \mathcal{E}_{\mathsf{sym}})$ to $\mathcal{B}$. The unknown master secret key $\mathsf{MSK}_{\mathsf{SM9}}$ is γ. $\mathcal{B}$ selects a collision resistant hash function H and a pseudo-random function PRF. Further, $\mathcal{B}$ chooses a random $\mathsf{ssk}_1 \in \{0,1\}^{l_1}$ and sets $\mathsf{SSK}_1 = (\mathsf{ssk}_1, \mathsf{ctr} = 1)$. Finally, $\mathcal{B}$ sends the master public key $\mathsf{MPK} = (\mathcal{G}, g, h, u, w, F, H, \mathrm{PRF}, \mathcal{E}_{\mathsf{sym}})$, and the initial server secret key SSK_1 to $\mathcal{A}$. The simulator $\mathcal{B}$ updates the server secret key periodically and it will send all updated server secret keys to $\mathcal{A}$.

Phase 1: $\mathcal{A}$ adaptively issues queries to following three oracles:

- $\mathcal{O}_{\mathsf{Extract}}(\mathsf{ID}_i)$: $\mathcal{B}$ sends the query to $\mathcal{O}^{\mathsf{SM9}}_{\mathsf{Extract}}(\mathsf{ID}_i)$ and forwards the obtained secret key $\mathsf{SK}_{\mathsf{ID}_i}$ to $\mathcal{A}$.
- $\mathcal{O}_{\mathsf{Dec}}(\mathsf{ID}_i, \mathsf{CT}_{\mathsf{ID}_i,\mathsf{ctr}})$: $\mathcal{B}$ runs $\mathsf{Transform}(\mathsf{MPK}, \mathsf{CT}_{\mathsf{ID}_i,\mathsf{ctr}}, \mathsf{SSK}_{\mathsf{ctr}}, \emptyset)$ to obtain $\mathsf{TCT}_{\mathsf{ID}_i}$. It then issues a query to $\mathcal{O}^{\mathsf{SM9}}_{\mathsf{Dec}}$ on $(\mathsf{ID}_i, \mathsf{TCT}_{\mathsf{ID}_i})$. $\mathcal{C}$ returns a plaintext. $\mathcal{B}$ forwards the plaintext to $\mathcal{A}$.
- $\mathcal{O}_{\mathsf{SSK}}(\mathsf{ctr}_i)$: If the server secret key $\mathsf{SSK}_{\mathsf{ctr}_i}$ has not been generated, $\mathcal{B}$ randomly samples $\mathsf{SSK}_{\mathsf{ctr}_i} \in \{0,1\}^{l_1}$ and stores $\mathsf{SSK}_{\mathsf{ctr}_i}$. Otherwise, $\mathcal{C}$ recalls $\mathsf{SSK}_{\mathsf{ctr}_i}$ from its storage. $\mathcal{B}$ outputs $\mathsf{SSK}_{\mathsf{ctr}_i}$ to $\mathcal{A}$.

Challenge: $\mathcal{A}$ outputs a challenge identity ID^*, a challenge time counter ctr^* and two equal length messages $m_0, m_1 \in \{0,1\}^*$. Note that ID^* cannot appear in previous queries to $\mathcal{O}_{\mathsf{Extract}}$. If $\mathsf{SSK}_{\mathsf{ctr}^*}$ has not been generated, $\mathcal{B}$ randomly samples $\mathsf{SSK}_{\mathsf{ctr}^*}$. $\mathcal{B}$ forwards ID^* and m_0, m_1 to $\mathcal{C}$. On receiving the challenge, $\mathcal{C}$ chooses a fair coin $\mu \in \{0,1\}$, runs $\mathsf{Enc}_{\mathsf{SM9}}(\mathsf{MPK}_{\mathsf{SM9}}, \mathsf{ID}_i, m_\mu)$ to obtain $\mathsf{CT}^*_{\mathsf{ID}^*}$ and sends $\mathsf{CT}^*_{\mathsf{ID}^*}$ to $\mathcal{B}$. Finally, $\mathcal{B}$ runs $\mathsf{CTInit}(\mathsf{MPK}, \mathsf{CT}^*_{\mathsf{ID}^*}, \mathsf{SSK}_{\mathsf{ctr}^*})$ to obtain $\mathsf{CT}^*_{\mathsf{ID}^*,\mathsf{ctr}^*}$, and sends $\mathsf{CT}^*_{\mathsf{ID}^*,\mathsf{ctr}^*}$ to $\mathcal{A}$.

Phase 2: $\mathcal{A}$ adaptively issues more queries to following three oracles:

- $\mathcal{O}_{\mathsf{Extract}}(\mathsf{ID}_i)$ where $\mathsf{ID}_i \neq \mathsf{ID}^*$: $\mathcal{B}$ responds as in **Phase 1**.
- $\mathcal{O}_{\mathsf{Dec}}(\mathsf{ID}_i, \mathsf{CT}_{\mathsf{ID}_i,\mathsf{ctr}})$ where $(\mathsf{ID}_i, \mathsf{CT}_{\mathsf{ID}_i,\mathsf{ctr}}) \neq (\mathsf{ID}^*, \mathsf{CT}^*_{\mathsf{ID}^*,\mathsf{ctr}^*})$: $\mathcal{B}$ responds as in **Phase 1**.
- $\mathcal{O}_{\mathsf{SSK}}(\mathsf{ctr}_i)$: $\mathcal{B}$ responds as in **Phase 1**.

Guess: $\mathcal{A}$ outputs a guess μ' of μ. $\mathcal{B}$ forwards μ' to $\mathcal{C}$.

As shown above, the master public key, the server secret key, secret keys and ciphertexts generated by $\mathcal{B}$ is of identical distribution to those of the proposed SA-IR-RIBE scheme. If $\mathcal{A}$ successfully guesses which message is encrypted in the challenge ciphertext, $\mathcal{B}$ also outputs the right guess. Therefore, if $\mathcal{A}$ can break the proposed SA-IR-RIBE scheme with probability $\epsilon(\lambda)$, $\mathcal{B}$ can break SM9-IBE with the same probability. This completes the proof of Theorem 1. □

Theorem 2 *The proposed server-aided Identity-Based Encryption scheme is adaptive-ID CPA-secure for User Revocation Validity with respect to Definition 2.*

The Theorem 2 is proved briefly as follows.

Proof Let ID^* be the identity of the revoked user. Let ctr^* be the challenge time counter and $\mathsf{CT}^*_{\mathsf{ID}^*,\mathsf{ctr}^*}$ be the challenge ciphertext. The adversary $\mathcal{A}$ (the revoked user) holds the secret key $\mathsf{SK}_{\mathsf{ID}^*}$, and queries on $\mathcal{O}_{\mathsf{Extract}}(\mathsf{ID}_i)$ and $\mathcal{O}_{\mathsf{SSK}}(\mathsf{ctr}_i)$ with the constraint that $\mathsf{ctr}_i \neq \mathsf{ctr}^*$.

In the **Challenge** phase, $\mathcal{A}$ receives the challenge ciphertext $\mathsf{CT}^*_{\mathsf{ID}^*,\mathsf{ctr}^*} = ((C_1^*)^\Delta, C_2^*)$ where $\Delta = \mathrm{PRF}(\mathsf{SSK}_{\mathsf{ctr}^*}, H(C_2^*))$. In this case, $e(\mathsf{SK}_{\mathsf{ID}^*}, (C_1^*)^\Delta) = e(g,h)^{\gamma z \Delta}$. Let $t = e(g,h)^{\gamma z}$, which is required to derive the encapsulated key K. Provided that Δ is randomly generated by the PRF, $(e(g,h)^{\gamma z \Delta}, t)$ can be viewed as an ElGamal KEM instance $(u^{z\Delta}, u^z)$, where $u = e(g,h)^\gamma$. Therefore, if the adversary $\mathcal{A}$ can break the User Revocation Validity security of the proposed scheme, we can build a simulator $\mathcal{B}$ to break the security of ElGamal KEM. Thus, Theorem 2 is proved. □

Performance evaluation

In this section, we give thorough performance analyses of the proposed scheme.

Theoretical Analysis. As shown in Table 2, we compare our revocation mechanism with many existing works (Boneh and Franklin 2001; Boldyreva et al. 2008; Libert and Vergnaud 2009; Seo and Emura 2013; Li et al. 2013; Qin et al. 2015) in terms of functionality, security model, communication costs and computation costs. Note that we do not include schemes with impractical assumptions, e.g., (Boneh et al. 2001; Libert and Quisquater 2003). The comparison is performed on the symmetric pairing setting: $\mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$. Meanwhile, only operations of the KEM part are measured.

The works (Boneh and Franklin 2001; Boldyreva et al. 2008; Libert and Vergnaud 2009; Seo and Emura 2013; Qin et al. 2015) provide indirect revocation of users' secret keys meaning that the revocation needs a period of time to take effect. Our scheme and (Li et al. 2013) achieve direct user revocation by adopting a semi-trusted server. In our proposed SA-IR-RIBE scheme, the sizes of master public key, secret key and ciphertext are all constant, i.e., $O(1)$. Libert and Vergnaud (2009); Seo and Emura (2013); Qin et al. (2015) adopted Waters IBE (Waters 2005) as the underlying IBE scheme thus had long master public keys, i.e., $O(\log N)$. (Boldyreva et al. 2008; Libert and Vergnaud 2009; Seo and Emura 2013) followed the BGK approach (Boldyreva et al. 2008) resulting long secret keys, i.e., $O(\log N)$. Compared with all listed schemes (Boneh and Franklin 2001; Boldyreva et al. 2008; Libert and Vergnaud

Table 2 Comparisons with revocable IBE schemes

	(Boneh and Franklin 2001)	(Boldyreva et al. 2008)	(Libert and Vergnaud 2009)	(Seo and Emura 2013)	(Li et al. 2013)	(Qin et al. 2015)	Ours
Revocation Mode	Indirect	Indirect	Indirect	Indirect	Direct	Indirect	Direct
Server	-	-	-	-	Semi-trusted	Untrusted	Semi-trusted
Master Public Key Size	$2\|\mathbb{G}\|$	$6\|\mathbb{G}\|$	$(6+\log N)\|\mathbb{G}\|$	$(6+\log N)\|\mathbb{G}\|$	$3\|\mathbb{G}\|$	$(6+\log N)\|\mathbb{G}\|$	$4\|\mathbb{G}\|$
Secret Key Size	$\|\mathbb{G}\|$	$2\log N\|\mathbb{G}\|$	$\log N(2\|\mathbb{G}\|+\|\mathbb{Z}_p\|)$	$2\log N\|\mathbb{G}\|$	$4\|\mathbb{G}\|$	$2\|\mathbb{G}\|$	$\|\mathbb{G}\|$
Ciphertext Size	$\|\mathbb{G}\|$	$3\|\mathbb{G}\|+\|\mathbb{G}_T\|$	$3\|\mathbb{G}\|+2\|\mathbb{G}_T\|$	$3\|\mathbb{G}\|+\|\mathbb{G}_T\|$	$3\|\mathbb{G}\|$	$3\|\mathbb{G}\|+\|\mathbb{G}_T\|$	$\|\mathbb{G}\|$
Key Update Size	$(N-r)\|\mathbb{G}\|$	$2r\log\frac{N}{r}\|\mathbb{G}\|$	$r\log\frac{N}{r}(2\|\mathbb{G}\|+\|\mathbb{Z}_p\|)$	$2r\log\frac{N}{r}\|\mathbb{G}\|$	$2(N-r)\|\mathbb{G}\|$	$2r\log\frac{N}{r}\|\mathbb{G}\|$	0
Key Generation Cost	1Exp	12Exp	4log *N*Exp	3log *N*Exp	6Exp	3Exp	1Exp
Encryption Cost	2Exp + 1P	7 log *N*Exp	5Exp + 2P	5Exp	3Exp	5Exp	3Exp
Decryption Cost	1Exp + 1P	2Exp + 4P	1Exp + 3P	3P	4P	3P	1P
Key Update Cost	$(N-r)$Exp	$7r\log\frac{N}{r}$Exp	$4r\log\frac{N}{r}$Exp	$3r\log\frac{N}{r}$Exp	$3(N-r)$Exp	$3r\log\frac{N}{r}$Exp	nExp

‡Exp and P denote a module exponentiation and a pairing computation, respectively. *N*, *r* and *n* indicate the numbers of users, revoked users and ciphertexts stored in the server, respectively

2009; Seo and Emura 2013; Li et al. 2013; Qin et al. 2015), the key generation, encryption, and decryption algorithms in our scheme are efficient. The key generation algorithm takes only 1 exponentiation, which is better than (Boldyreva et al. 2008; Libert and Vergnaud 2009; Seo and Emura 2013; Li et al. 2013; Qin et al. 2015) and equal with (Boneh and Franklin 2001). The encryption algorithm takes 3 exponentiations, which is better than (Boneh and Franklin 2001; Boldyreva et al. 2008; Libert and Vergnaud 2009; Seo and Emura 2013; Qin et al. 2015) and equal with (Li et al. 2013). The decryption algorithm costs only 1 pairing, which is better than all listed works (Boneh and Franklin 2001; Boldyreva et al. 2008; Libert and Vergnaud 2009; Seo and Emura 2013; Li et al. 2013; Qin et al. 2015). When a user is revoked, (Boneh and Franklin 2001; Boldyreva et al. 2008; Libert and Vergnaud 2009; Seo and Emura 2013; Li et al. 2013; Qin et al. 2015) had large communication costs and computation costs for key update. In contrast, our scheme does not have communication costs and only performs ciphertext updates on the server side. Above theoretical discussions show the advantage and efficiency of the proposed scheme.

Experimental Analysis. To validate the reality of our ideas, we instantiate and implement the proposed scheme following the standard (Gm/t 2016b). The primitives EC2OSP, FE2OSP, KDF2 and $\mathrm{H2RF}_i$ are constructed as Shoup (2006); Cheng (2017). The bilinear pairing is the standard 256-bit BN curve as specified in Gm/t (2016b). The hash functions H_v and H are implemented using the SM3 hash algorithm (Wang and Yu 2016), which takes input a message m of length l ($l < 2^{64}$), and outputs a 256-bit hash. Thus, $l_1 = l_2 = v = 256$. The PRF is instantiated as follows:

$$\mathrm{PRF}(a,b) = \text{HMAC-SM3}(a,b) = \mathrm{SM3}((a \oplus opad)||\mathrm{SM3}((a \oplus ipad)||b))$$

where *ipad* is the byte $0x36$ repeated 64 times and *opad* is the byte $0x5C$ repeated 64 times. For the symmetric encryption primitive $\mathcal{E}_{\mathsf{Sym}}$, we use the DEM construction in Gm/t (2016b), where the message is encrypted using XOR encryption. Concretely, the DEM part is computed as follows:

$$K_1||K_2 = \mathrm{KDF2}(H_v, \mathrm{EC2OSP}(C_1)||\mathrm{FE2OSP}(t)||\mathrm{ID}_i, |m|+v),$$
$$C' = K_1 \oplus m, \quad C_2 = (C', H_v(C'||K_2)).$$

We implement the proposed SA-IR-RIBE scheme on top of the GmSSL library [1]. We conduct comprehensive benchmark on one core of an Intel Xeon CPU E5-2609 v4 @1.70 GHz and 128GB RAM running Ubuntu 16.04 LTS 64-bit. Since the network speed changes according to the user, only the running time of each algorithm is measured. The size of the message is set to 75 KB, which is a typical size of an email file[2]. Each algorithm is executed 100 times. The average running results are taken as the final results. The detailed result is shown in Table 3. It takes

[1] http://gmssl.org/

[2] https://www.lifewire.com/what-is-the-average-size-of-an-email-message-1171208

Table 3 Performance evaluation of the proposed mechanism

Algorithm	Setup	Extract	Enc	CTInit	SSKUpdate	CTUpdate	Transform	Dec
Time (ms)	0.76	26.99	405.68	1.73	6.45×10^{-5}	3.36	1.78	351.34

405.68 ms and 351.34 ms for user-side message encryption and decryption, while algorithms CTInit, SSKUpdate, CTUpdate running on the server only cost 1.73 ms, 6.45×10^{-5} ms and 3.36 ms, respectively.

As our revocation scheme adds additional overhead on the server side, we further measure performance of the server with the following procedures: 1) On time ctr_1, N users in the system encrypt their messages and send the encrypted payloads to the server. Upon receiving a user's upload request, the server runs the algorithm CTInit with its server secret key $\mathsf{SSK}_{\mathsf{ctr}_1}$ to generate the new ciphertext $\mathsf{CT}_{\mathsf{ID}_i,\mathsf{ctr}_1}$, and stores it on the disk. 2) When updated to time ctr_2, the server first updates its server secret key $\mathsf{SSK}_{\mathsf{ctr}_1}$ to $\mathsf{SSK}_{\mathsf{ctr}_2}$ with algorithm SSKUpdate, then runs CTUpdate to update all stored ciphertexts. 3) N valid users send download requests to the server, the server runs Transform with $\mathsf{SSK}_{\mathsf{ctr}_2}$ and sends the transformed ciphertexts to the users. Note that for simplicity, we do not consider network latency and disk I/O time in this benchmark, so only the server processing time is measured. We set the message size sent by each user to 75 KB, and increase N from 0 to 10,000. To fully leverage server's computational power, we integrate OpenMP (Dagum and Menon 1998) to support parallel executions and compare with non-parallel ones.

In Fig. 4 and Table 4, we present the experiment result. The integration of OpenMP has significantly reduced latency and improved throughput of the server. With OpenMP disabled, the processing time is about tens of seconds. The algorithm CTInit takes 18.43 seconds to initialize 10,000 users' uploaded messages. The algorithm CTUpdate takes 35.57 seconds to update all the stored ciphertexts. The algorithm Transform costs 19.56 seconds to transform the stored ciphertexts for 10,000 users' requests. With OpenMP enabled, the obtained performance is satisfactory. CTInit only takes 2.91 seconds to initialize 10,000 users' uploaded messages. CTUpdate takes 4.85 seconds to update all the stored ciphertexts. Transform costs 2.68 seconds to transform the stored ciphertexts for 10,000 users' requests. During the experiment, algorithms CTInit, SSKUpdate, CTUpdate reach throughput of 269 MB/s, 151 MB/s and 277 MB/s respectively. The statistics show that the proposed scheme is practical for real-world systems.

Conclusion

In this paper, focusing on the problem of SM9-IBE revocation, we proposed a server-aided revocation mechanism. Our model has three desirable properties: (1) Compact system model: an existing server is adopted to perform all heavy workloads during user revocation; (2) Efficient and immediate revocation: the revocation takes effect immediately by taking simple operations in the server; and (3) Key-exposure resistance: even if the server secret key is leaked, a revoked user still cannot decrypt any data. We

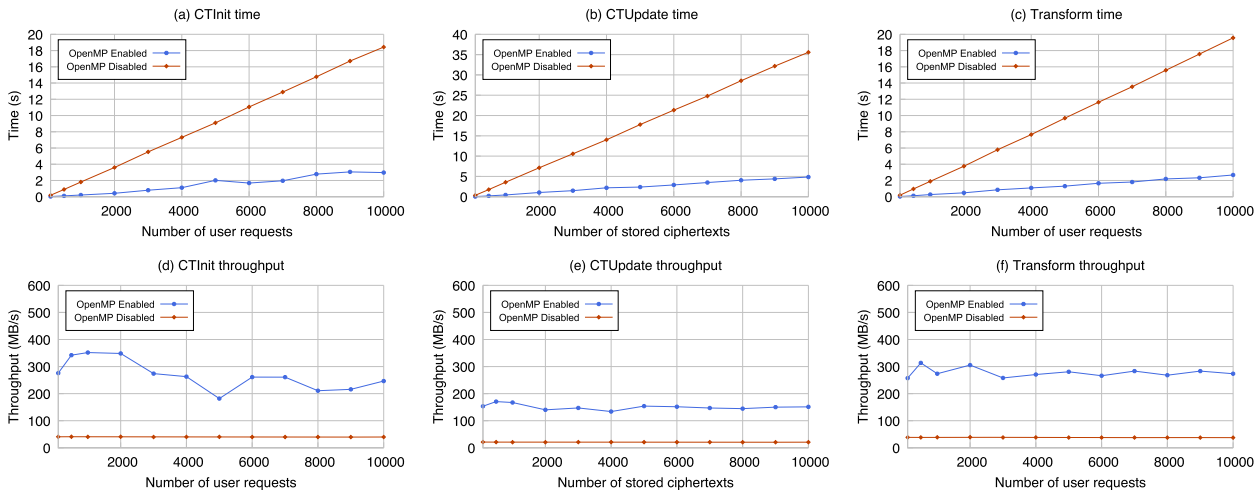

Fig. 4 Performance evaluation of the server

Table 4 Performance evaluation of the server

#Users	OpenMP*	CTInit		CTUpdate		Transform	
		Time	Throughput	Time	Throughput	Time	Throughput
100	N	0.18	40.63	0.34	21.11	0.19	38.46
	Y	0.03	275.99	0.05	153.42	0.03	257.31
500	N	0.90	40.90	1.76	20.89	0.96	38.37
	Y	0.11	342.12	0.21	170.68	0.12	313.51
1,000	N	1.81	40.43	3.57	20.53	1.90	38.70
	Y	0.21	351.81	0.44	167.30	0.27	273.58
2,000	N	3.60	40.76	7.14	20.56	3.76	39.06
	Y	0.42	348.42	1.05	140.09	0.48	305.55
3,000	N	5.53	39.82	10.57	20.83	5.79	38.03
	Y	0.80	273.93	1.50	147.24	0.85	258.03
4,000	N	7.31	40.16	14.02	20.92	7.65	38.33
	Y	1.12	262.84	2.19	133.82	1.08	270.54
5,000	N	9.10	40.32	17.78	20.63	9.68	37.90
	Y	2.02	181.82	2.38	154.01	1.31	280.72
6,000	N	11.05	39.81	21.33	20.64	11.64	37.82
	Y	1.69	261.18	2.90	151.68	1.65	266.10
7,000	N	12.89	39.82	24.80	20.70	13.55	37.90
	Y	1.97	260.82	3.50	146.89	1.81	283.20
8,000	N	14.77	39.73	28.57	20.54	15.57	37.68
	Y	2.78	210.70	4.06	144.58	2.19	268.05
9,000	N	16.71	39.51	32.16	20.53	17.57	37.57
	Y	3.06	215.96	4.40	150.07	2.33	283.38
10,000	N	18.43	39.81	35.57	20.62	19.57	37.49
	Y	2.98	246.49	4.85	151.20	2.68	273.60

‡Metrics are collected on an Ubuntu 16.04 server with 8 Intel E5-2609 v4 @1.70 GHz cores. Note that time in the table above is given in seconds (s). Throughputs are measured by evaluating the average size of data (in megabytes) processed per second (MB/s)
*In this column, Y means OpenMP is enabled where N means OpenMP is not enabled

further prove that the proposed mechanism is adaptive-ID chosen ciphertext secure. Finally, we present both theoretical and experimental analyses showing that the proposed mechanism is practical and efficient.

Acknowledgements
The authors would like to thank the anonymous reviewers for their valuable comments.

Authors' contributions
Shuzhou Sun and Hui Ma proposed the revocation mechanism for SM9, and drafted the manuscript. Rui Zhang participated in problem discussions and improvements of the manuscript. Wenhan Xu implemented and benchmarked the proposed scheme. All authors read and approved the final manuscript.

Funding
This work was partially supported by National Natural Science Foundation of China (Nos. 61772520, 61802392, 61972094, 61472416, 61632020), Key Research and Development Project of Zhejiang Province (Nos. 2017C01062, 2020C01078), Beijing Municipal Science & Technology Commission (Project Number. Z191100007119007, Z191100007119002).

Availability of data and materials
Not applicable.

Competing interests
The authors declare that they have no competing interests.

Author details
[1]State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, China. [2]School of Cyber Security, University of Chinese Academy of Sciences, Beijing 100049, China.

Received: 3 September 2019 Accepted: 26 April 2020
Published online: 13 May 2020

References

Bentahar K, Farshim P, Malone-Lee J, Smart NP (2008) Generic constructions of identity-based and certificateless kems. J Cryptol 21(2):178–199
Boldyreva A, Goyal V, Kumar V (2008) Identity-based encryption with efficient revocation. In: Proceedings of the 15th ACM Conference on Computer and Communications Security. ACM. pp 417–426. https://doi.org/10.1145/1455770.1455823

Boneh D, Boyen X (2004) Efficient selective-id secure identity-based encryption without random oracles. In: International Conference on the Theory and Applications of Cryptographic Techniques. Springer. pp 223–238. https://doi.org/10.1007/978-3-540-24676-3_14
Boneh D, Boyen X (2004) Secure identity based encryption without random oracles. In: Annual International Cryptology Conference. Springer. pp 443–459. https://doi.org/10.1007/978-3-540-28628-8_27
Boneh D, Ding X, Tsudik G, Wong C-M (2001) A method for fast revocation of public key certificates and security capabilities. In: USENIX Security Symposium. p 22
Boneh D, Franklin M (2001) Identity-based encryption from the weil pairing. In: Annual International Cryptology Conference. Springer. pp 213–229. https://doi.org/10.1137/s0097539701398521
Boyen X (2007) General ad hoc encryption from exponent inversion ibe. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. Springer. pp 394–411. https://doi.org/10.1007/978-3-540-72540-4_23
Boyen X, Martin L (2007) Identity-based cryptography standard (ibcs)# 1: Supersingular curve implementations of the bf and bb1 cryptosystems. Technical report, RFC 5091, December. https://doi.org/10.17487/rfc5091
Canetti R, Halevi S, Katz J (2004) Chosen-ciphertext security from identity-based encryption. In: International Conference on the Theory and Applications of Cryptographic Techniques. Springer. pp 207–222. https://doi.org/10.1007/978-3-540-24676-3_13
Cheng Z (2017) The SM9 Cryptographic Schemes. Cryptol ePrint Arch. Report 2017/117. https://eprint.iacr.org/2017/117
Cocks C (2001) An identity based encryption scheme based on quadratic residues. In: IMA International Conference on Cryptography and Coding. Springer. pp 360–363. https://doi.org/10.1007/3-540-45325-3_32
Dagum L, Menon R (1998) Openmp: An industry-standard api for shared-memory programming. Comput Sci Eng 1:46–55
Delerablée C (2007) Identity-based broadcast encryption with constant size ciphertexts and private keys. In: International Conference on the Theory and Application of Cryptology and Information Security. Springer. pp 200–215. https://doi.org/10.1007/978-3-540-76900-2_12
Ge A, Wei P (2019) Identity-based broadcast encryption with efficient revocation. In: IACR International Workshop on Public Key Cryptography. Springer. pp 405–435. https://doi.org/10.1007/978-3-030-17253-4_14
Gentry C (2006) Practical identity-based encryption without random oracles. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. Springer. pp 445–464. https://doi.org/10.1007/11761679_27
Gm/t (2016) 0044.1-2016 identity-based cryptographic algorithms sm9-part 1: General. Technical report
Gm/t (2016) 0044.5-2016 identity-based cryptographic algorithms sm9-part 5: Parameter definition. Technical report
IEEE (2013) std 1363.3-2013 ieee standard for identity-based cryptographic techniques using pairings. Technical report. IEEE Comput Soc
Iso/iec (2015) 18033-5:2015 information technology – security techniques – encryption algorithms – part 5: Identity-based ciphers. Technical report, ISO/IEC
Kogan N, Shavitt Y, Wool A (2006) A practical revocation scheme for broadcast encryption using smartcards. ACM Trans Inf Syst Secur (TISSEC) 9(3):325–351
Li J, Jia C, Li J, Chen X (2012) Outsourcing encryption of attribute-based encryption with mapreduce. In: International Conference on Information and Communications Security. Springer. pp 191–201. https://doi.org/10.1007/978-3-642-34129-8_17
Li J, Li J, Chen X, Jia C, Lou W (2013) Identity-based encryption with outsourced revocation in cloud computing. IEEE Trans Comput 64(2):425–437
Libert B, Quisquater J-J (2003) Efficient revocation and threshold pairing based cryptosystems. In: Proceedings of the Twenty-second Annual Symposium on Principles of Distributed Computing. ACM. pp 163–171. https://doi.org/10.1145/872035.872059
Libert B, Vergnaud D (2009) Adaptive-id secure revocable identity-based encryption. In: Cryptographers' Track at the RSA Conference. Springer. pp 1–15. https://doi.org/10.1007/978-3-642-00862-7_1
Martin L, Appenzeller G, Schertler M (2009) Identity-based encryption architecture and supporting data structures. Identity. https://doi.org/10.17487/rfc5408
Martin L, Schertler M (2009) Using the boneh-franklin and boneh-boyen identity-based encryption algorithms with the cryptographic message syntax (cms). https://doi.org/10.17487/rfc5409
Naor D, Naor M, Lotspiech J (2001) Revocation and tracing schemes for stateless receivers. In: Annual International Cryptology Conference. Springer. pp 41–62. https://doi.org/10.1007/3-540-44647-8_3
Qin B, Deng RH, Li Y, Liu S (2015) Server-aided revocable identity-based encryption. In: European Symposium on Research in Computer Security. Springer. pp 286–304. https://doi.org/10.1007/978-3-319-24174-6_15
Sahai A, Waters B (2005) Fuzzy identity-based encryption. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. Springer. pp 457–473. https://doi.org/10.1007/11426639_27
Sakai R, Kasahara M (2003) Id based cryptosystems with pairing on elliptic curve. IACR Cryptol ePrint Arch 2003:54
Seo JH, Emura K (2013) Revocable identity-based encryption revisited: Security model and construction. In: International Workshop on Public Key Cryptography. Springer. pp 216–234. https://doi.org/10.1007/978-3-642-36362-7_14
Shoup V (2001) A proposal for an iso standard for public key encryption (version 2.1). IACR e-Print Arch 112
Shoup V (2006) ISO/IEC 18033-2: 2006. Information Technology–Security Techniques–Encryption Algorithms–Part 2
Wang X, Yu H (2016) Sm3 cryptographic hash algorithm. J Inf Secur Res:983–994
Waters B (2005) Efficient identity-based encryption without random oracles. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. Springer. pp 114–127. https://doi.org/10.1007/11426639_7

Publisher's Note