Server-Aided Revocable Attribute-Based Encryption

Hui Cui[1](✉), Robert H. Deng[1], Yingjiu Li[1], and Baodong Qin[2]

[1] School of Information Systems, Secure Mobile Centre,
Singapore Management University, Singapore, Singapore
{hcui,robertdeng,yjli}@smu.edu.sg
[2] School of Computer Science and Technology,
Southwest University of Science and Technology, Mianyang, China
qinbaodong@swust.edu.cn

Abstract. As a one-to-many public key encryption system, attribute-based encryption (ABE) enables scalable access control over encrypted data in cloud storage services. However, efficient user revocation has been a very challenging problem in ABE. To address this issue, Boldyreva, Goyal and Kumar [5] introduced a revocation method by combining the binary tree data structure with fuzzy identity-based encryption, in which a key generation center (KGC) periodically broadcasts key update information to all data users over a public channel. The Boldyreva-Goyal-Kumar approach reduces the size of key updates from linear to logarithm in the number of users, and it has been widely used in subsequent revocable ABE systems; however, it requires each data user to keep a private key of logarithmic size and all non-revoked data users to periodically update decryption keys for each new time period. To further optimize user revocation in ABE, in this paper, we propose a notion called server-aided revocable ABE (SR-ABE), in which almost all workloads of data users incurred by user revocation are delegated to an untrusted server and each data user only needs to store a key of constant size. We then define a security model for SR-ABE, and present a concrete SR-ABE scheme secure under this model. Interestingly, due to the key embedding gadget employed in the construction of SR-ABE, our SR-ABE scheme does not require any secure channels for key transmission, and also enjoys an additional property in the decryption phase, where a data user only needs to perform one exponentiation computation to decrypt a ciphertext.

Keywords: Revocation · Attribute-based encryption · Server-aided

1 Introduction

Attribute-based encryption (ABE) [22] is a promising solution to preserve data privacy in scenarios where data users are identified by their attributes (or credentials) and data owners want to share their data stored in the cloud with data users

I. Askoxylakis et al. (Eds.): ESORICS 2016, Part II, LNCS 9879, pp. 570–587, 2016.
DOI: 10.1007/978-3-319-45741-3_29

whose attributes satisfy a certain access structure (or policy). In a ciphertext-policy ABE (CP-ABE) system, a trusted key generation center (KGC) issues a private key for every data user corresponding to his/her attribute set, and each data owner specifies an access policy over an attribute set to an encrypted message[1]. A data user is able to decrypt a ciphertext if the attribute set associated with his/her private key satisfies the access policy ascribed to the ciphertext.

Since an ABE system may involve a large number of data users, efficient user revocation, due to either private key compromises or user resignations, has been regarded as a very important and challenging problem. Boldyreva, Goyal and Kumar [5] put forth an efficient revocation method by combining the fuzzy identity-based encryption (IBE) scheme [22] with the binary tree data structure [18], where the KGC issues a long-term private key to each data user and publicly broadcasts key updates at the beginning of each time period, but only non-revoked data users can generate decryption keys from their long-term private keys and the key updates to decrypt the newly created ciphertexts. The revocable ABE schemes in [1,5,9,21] following the Boldyreva-Goyal-Kumar approach mitigate the KGC's communication overhead incurred in the key update process, but they fail to reduce the workloads of data users since every data user is required to keep a private key of logarithmic size and all non-revoked data users need to periodically update decryption keys to decrypt newly encrypted data. Regarding this crux, Qin et al. [19] proposed a solution in identity-based encryption, called server-aided revocable identity-based encryption (SR-IBE), where almost all workloads on data users are delegated to a untrusted server who manages data users' public keys and key updates sent by the KGC periodically, and each data user keeps just one private key of constant size (i.e., $O(1)$) and are not required to communicate with either the KGC or the untrusted server during the key update phase. However, this problem has not caught sufficient attention in the attribute-based setting.

1.1 Our Contributions

Motivated by SR-IBE in [19], we put forth a notion called server-aided revocable ABE (SR-ABE) to accomplish efficient and secure user revocation in ABE. The architecture of an SR-ABE scheme is depicted in Fig. 1 under the scenario of cloud storage [24]. The architecture consists of four types of entities: a KGC, data owners, data users and an untrusted server[2]. Note that the untrusted server could be operated by anyone, including the cloud storage system. The KGC possesses a master private key, and publishes its public parameter. When a new data user,

[1] There are two complimentary forms of ABE: CP-ABE and key-policy ABE (KP-ABE). In a KP-ABE system, the situation is reversed in that a private key is associated with an access policy and a ciphertext is associated with a set of attributes. In the rest of the paper, unless otherwise specified, we will focus on CP-ABE.

[2] The server is untrusted in the sense that it honestly follows the protocol, but does not hold any secret information (i.e., it may collude with data users), and all operations done by the server can be performed by anyone, including data users (i.e., any dishonest behaviour from the server can be easily detected).

say Alice, joins the system, she first generates a public and private user-key pair by herself. She keeps the private user-key to herself and sends the public user-key (along with a proof showing that she knows the corresponding private user-key) to the KGC, which, based on Alice's public user-key and attributes, generates a public attribute-key for Alice and sends it to the untrusted server. Also, the KGC periodically generates key updates for all non-revoked data users and publicly transmits them to the untrusted server. The same as that in the standard CP-ABE, to upload a message in the current time period to the cloud, a data owner encrypts the message over an access structure and a time period using the system public parameter, and outsources the resulting ciphertext to the cloud. To decrypt a ciphertext, a data user forwards the ciphertext to the untrusted server. If the data user is not revoked and his/her set of attributes satisfies the access structure ascribed to the ciphertext, the untrusted server is able to generate a transformation key from his/her public attribute-key and the key update information, with which the server can partially decrypt the ciphertext. This partially decrypted ciphertext can be fully decrypted by the data user using his/her private user-key. Notice that SR-ABE only requires all data users to contact the KGC during the user registration phase, while operations caused by user revocation are completely handled by the untrusted server and are totally transparent to the data users.

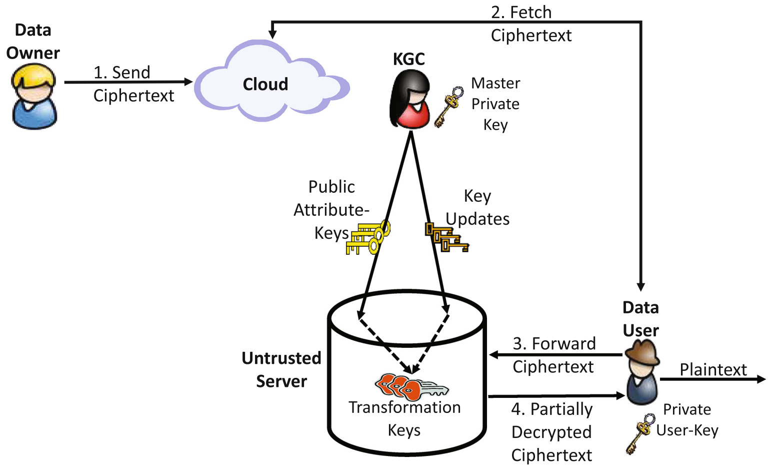

Fig. 1. System architecture of server-aided revocable attribute-based encryption.

The key challenge in constructing an SR-ABE scheme is how to enable the untrusted server to assist decryption while without knowing the underlying plaintext. In an IBE system, each user has a unique identity and every ciphertext is exclusively designated to one recipient. Therefore, in the SR-IBE scheme presented in [19], after the server partially decrypts a ciphertext for a data user, the user can obtain the underlying plaintext using his/her identity-based private key. However, in an ABE scheme, the same attributes could be shared among multiple users, so if using the master private key splitting methodology in [19] in an SR-ABE scheme, given the partial decryption of a ciphertext by the untrusted server, a data user would be able to fully decrypt the partially decrypted ciphertext, regardless of the data user being revoked or not, as long as his/her set of attributes satisfies the access structure in the ciphertext. To conquer this challenge, we equip each data user with a pair of self-generated public and private user-keys[3] (i.e., it does not require a secure channel for key transmission), and then trickly embed the public user-key into the public attribute-key generated by the KGC. As a result, the untrusted server can still partially decrypt ciphertexts for non-revoked users, but every partially decrypted ciphertext is bound with a public user-key, which can only be decrypted by the user possessing the corresponding private user-key.

We define a security model for SR-ABE, which formalizes the possible realistic threats and takes into account all adversarial capabilities of the standard ABE security notion. The adversary is able to learn private user-keys and public attribute-keys of data users with attributes of its choice. The adversary should not be able to learn any partial information about the message encrypted for the challenge access structure. In addition, we consider the adversary having access to periodic key updates, transformation keys for different time periods and being able to revoke users of its choice. The adversary should also not be able to learn any partial information about the messages encrypted for any revoked data user whose attributes satisfy the challenge access structure when the encryption is done after the time of revocation.

Then we present a concrete SR-ABE construction for this model based on the large universe CP-ABE scheme in the prime-order groups presented by Rouselakis and Waters [20]. For the sake of building the SR-ABE scheme, we resort to the technique in [23] and the binary tree data structure [18], and combine them with the Rouselakis-Waters CP-ABE scheme [20]. In our SR-ABE scheme, components corresponding to each attribute in a transformation key follow the form of the second level private key of the HIBE scheme [6]. A technique similar to that in [23] is used to generate the public attribute-keys and key updates, where the master private key of the KGC is randomly divided into two parts and each part is respectively bound to the public attribute-keys and key updates. Also, to reduce the size of key updates from linear to logarithmic in the number of data users, the binary tree data structure in [18] is used. We present the full details of the construction in Sect. 4. It is worth noticing that though SR-ABE is derived from SR-IBE, due to the gadget we employ in the public attribute-key generation algorithm, our SR-ABE construction enjoys two additional advantages that the SR-IBE scheme in [19] does not have: (1) there is no need of secure channels for the distribution of private keys, since they are generated by each data user

[3] This user-key pair can also be generated and securely sent to the data user by the KGC as that in [19], but this requires a secure channel between the data user and the KGC for key distribution.

himself/herself; (2) in the decryption phase, each privileged data user only needs to perform one exponentiation computation and no pairing computation.

Since the Rouselakis-Waters CP-ABE scheme [20] is selectively secure, where the adversary has to commit the challenge access structure in advance, our SR-ABE scheme which is constructed based on [20] is also selectively secure. Note that the techniques can be applied to fully secure ABE schemes (e.g., [21]) to obtain fully secure server-aided ABE schemes.

In a nutshell, our contributions in this paper can be summarized as follows.

- We first propose a notion called server-aided revocable attribute-based encryption (SR-ABE), in which almost all data users' workloads incurred in key update phase are delegated to an untrusted server and each data user only needs to keep a private user-key of constant size for decryption.
- We define a security model for SR-ABE which considers all possible adversarial behaviours that could be executed by an adversary in the real world.
- Due to the gadget employed in the construction of SR-ABE, our SR-ABE scheme does not require any secure channels for key transmission, and enjoys an additional property in the decryption phase, where a data user only needs to perform one exponentiation computation to decrypt a ciphertext.

1.2 Related Work

Revocable IBE. Boneh and Franklin [8] suggested to renew users' private keys periodically to achieve user revocation in IBE, but this requires all users to regularly contact the KGC over secure channels, regardless of whether their keys have been exposed. That is, the size of key updates is linear in the number of non-revoked users (i.e., $O(N - R)$, where N is the number of all users and R is the number of revoked users). Hanaoka et al. [11] presented a method for users to periodically renew their private keys without interacting with the KGC, where the KGC publicly posts the key update information; however, each user needs to possess a tamper-resistant hardware device, making the solution rather cumbersome. Boldyreva, Goyal and Kumar [5] presented an efficient revocable IBE scheme to reduce the size of key updates from linear to logarithmic (i.e., $O(R \log(\frac{N}{R})))$ and remove the secure channels required during key updates, but all non-revoked users still need to periodically update their private keys for decryption. There are also revocable IBE schemes with a third party [3,7,10,14,16,17,19], where a semi-trusted[4] or untrusted third party is required to hold the shares of all users' private keys and help them decrypt. Once a user is revoked, the third party stops decrypting (or is disallowed to decrypt) for the user.

Revocable ABE. Two kinds of user revocation mechanisms have been proposed for revocable ABE [1,9]: direct and indirect revocation. In direct revocation, data owners directly specify the revocation list when encrypting [2,12,15]. In addition, Yang et al. [26] proposed a revocable ABE scheme by giving the direct

[4] In this paper, unless otherwise specified, "semi-trusted" means that the party is disallowed to collude with data users.

revocation capability to a semi-trusted server who shares the decryption ability with data users, and will terminate decryption operations for revoked users. In indirect revocation, the KGC indirectly disables revoked users through a key update process. Boldyreva, Goyal and Kumar [5] proposed a revocable KP-ABE scheme following the indirect revocation approach, Attrapadung and Imai [1] gave a hybrid revocable KP-ABE system which allows a data owner to select either direct or indirect revocation when encrypting a message, Sahai, Seyalioglu and Waters [21] provided a generic way to achieve indirect revocation in ABE schemes, and Cui and Deng [9] gave two revocable ABE schemes in the setting where the KGC'role is split across multiple KGCs.

Note that direct revocation can be done immediately without key updates, but it requires all data owners to keep a current revocation list. This makes the system impurely attribute-based, since data owners in the attribute-based setting create a ciphertext based solely on attributes without caring each data user's status. In this paper, we focus on ABE with indirect revocation.

1.3 Organization

The remainder of this paper is organized as follows. In Sect. 2, we briefly review the notions and definitions relevant to this paper. In Sect. 3, we describe the framework of our SR-ABE, and then present its security model. In Sect. 4, we give a concrete construction of SR-ABE, prove its security, and compare it with previous revocable ABE schemes. We conclude the paper in Sect. 5.

2 Preliminaries

In this section, we review the basic cryptographic definitions that are to be used in this paper.

2.1 Bilinear Pairings and Complexity Assumptions

Let G be a group of order p generated from g, and p be a prime number. We define $\hat{e} : G \times G \rightarrow G_1$ to be a bilinear map if it has the following properties [8].

- Bilinear: for all $g \in G$, and a, $b \in Z_p^*$, we have $\hat{e}(g^a, g^b) = \hat{e}(g, g)^{ab}$.
- Non-degenerate: $\hat{e}(g, g) \neq 1$.

We say that G is a bilinear group if the group operation in G is efficiently computable and there exists a group G_1 and an efficiently computable bilinear map $\hat{e} : G \times G \rightarrow G_1$ as above.

Decisional $(q-1)$ Assumption [20]. The decisional $(q-1)$ problem is that for any probabilistic polynomial-time algorithm, given $\overrightarrow{y} =$

$$\begin{array}{ll}
g, g^{\mu}, g^{1/a}, & \\
g^{a^i}, g^{b_j}, g^{\mu b_j}, g^{a^i b_j}, g^{a^i/b_j^2} & \forall\ (i,j) \in [q,q], \\
g^{a^i/b_j} & \forall\ (i,j) \in [2q,q] \text{ with } i \neq q+1, \\
g^{a^i b_j/b_{j'}^2} & \forall\ (i,j,j') \in [2q,q,q] \text{ with } j \neq j', \\
g^{\mu a^i b_j/b_{j'}}, g^{\mu a^i b_j/b_{j'}^2} & \forall\ (i,j,j') \in [q,q,q] \text{ with } j \neq j',
\end{array}$$

it is difficult to distinguish $(\overrightarrow{y}, \hat{e}(g,g)^{a^{q+1}\mu})$ from $(\overrightarrow{y}, Z)$, where $g \in G$, $Z \in G_1$, a, μ, b_1, ..., $b_q \in Z_p^*$ are chosen independently and uniformly at random.

2.2 Access Structures and Linear Secret Sharing

Definition 1 ***(Access Structure) [13,25].*** *Let $\{P_1, ..., P_n\}$ be a set of parties. A collection $\mathbb{A} \subseteq 2^{\{P_1,...,P_n\}}$ is monotone if $\forall B, C$: if $B \in \mathbb{A}$ and $B \subseteq C$, then $C \subseteq \mathbb{A}$. A monotone access structure is a monotone collection $\mathbb{A}$ of non-empty subsets of $\{P_1, ..., P_n\}$, i.e., $\mathbb{A} \subseteq 2^{\{P_1,...,P_n\}} \setminus \{\emptyset\}$. The sets in $\mathbb{A}$ are called authorized sets, and the sets not in $\mathbb{A}$ are called unauthorized sets.*

Definition 2 ***(Linear Secret Sharing Schemes (LSSS)) [13,25].*** *Let P be a set of parties. Let $\mathbb{M}$ be a matrix of size $l \times n$. Let $\rho : \{1, ..., l\} \rightarrow P$ be a function that maps a row to a party for labeling. A secret sharing scheme Π over a set of parties P is a linear secret-sharing scheme over Z_p if*

1. *The shares for each party form a vector over Z_p.*
2. *There exists a matrix $\mathbb{M}$ with l rows and n columns, called the share-generating matrix, for Π. For $x = 1, ..., l$, the x-th row of matrix $\mathbb{M}$ is labelled by a party $\rho(i)$, where $\rho : \{1, ..., l\} \rightarrow P$ is a function that maps a row to a party for labelling. Considering that the column vector $\overrightarrow{v} = (\mu, r_2, ..., r_n)$, where $\mu \in Z_p$ is the secret to be shared and $r_2, ..., r_n \in Z_p$ are randomly chosen, then $\mathbb{M}\overrightarrow{v}$ is the vector of l shares of the secret μ according to Π. The share $(\mathbb{M}\overrightarrow{v})_i$ belongs to party $\rho(i)$.*

It has been noted in [13] that every LSSS also enjoys the linear reconstruction property. Suppose that Π is an LSSS for an access structure $\mathbb{A}$. Let $\mathbf{A}$ be an authorized set, and define $I \subseteq \{1, ..., l\}$ as $I = \{i|\rho(i) \in \mathbf{A}\}$. Then the vector (1, 0, ..., 0) is in the span of rows of matrix $\mathbb{M}$ indexed by I, and there exist constants $\{w_i \in Z_p\}_{i \in I}$ such that, for any valid shares $\{v_i\}$ of a secret μ according to Π, we have $\sum_{i \in I} w_i v_i = \mu$. These constants $\{w_i\}$ can be found in polynomial time with respect to the size of the share-generating matrix $\mathbb{M}$ [4].

On the other hand, for an unauthorized set $\mathbf{A}'$, no such constants $\{w_i\}$ exist. Moreover, in this case it is also true that if $I' = \{i|\rho(i) \in \mathbf{A}'\}$, there exists a vector $\overrightarrow{w}$ such that its first component w_1 is any non-zero element in Z_p and $< \mathbb{M}_i, \overrightarrow{w} > = 0$ for all $i \in I'$, where $\mathbb{M}_i$ is the i-th row of $\mathbb{M}$ [20].

Boolean Formulas [13]. Access policies can also be described in terms of monotonic boolean formulas. LSSS access structures are more general, and can be derived from representations as boolean formulas. There are standard techniques to convert any monotonic boolean formula into a corresponding LSSS matrix. The boolean formula can be represented as an access tree, where the interior nodes are AND and OR gates, and the leaf nodes correspond to attributes. The number of rows in the corresponding LSSS matrix will be the same as the number of leaf nodes in the access tree.

2.3 Binary Tree

We recall the definition about binary tree described in [5,19]. Denote BT by a binary tree with N leaves corresponding to N users. Let **root** be the root node of the tree BT. If θ is a leaf node, then Path(θ) denotes the set of nodes on the path from θ to **root**, which includes both θ and **root**. If θ is a non-leaf node, then θ_l, θ_r denote left and right child of θ. Assume that nodes in the tree are uniquely encoded as strings, and the tree is defined by all of its node descriptions. The algorithm KUNodes is used to compute the minimal set of nodes for which key update needs to be published so that only the non-revoked users at a time period t are able to decrypt the ciphertexts. This algorithm takes a binary tree BT, a revocation list rl and a time period t as the input, and outputs a set of nodes which is the minimal set of nodes in BT such that none of the nodes in rl with corresponding time period before or at t (users revoked at or before t) have any ancestor (or, themselves) in the set, and all other leaf nodes (corresponding to non-revoked users) have exactly one ancestor (or, themselves) in the set. We give a pictorial depiction on how the KUNodes algorithm works in Fig. 2, where it firstly marks all the ancestors of the revoked nodes as revoked, and then it outputs all the non-revoked children of revoked nodes. Below is a

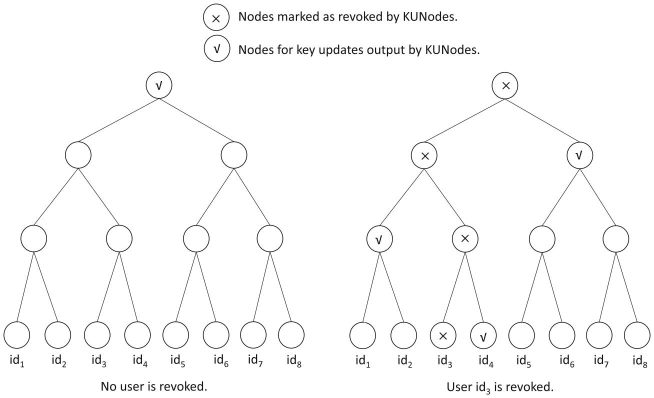
× Nodes marked as revoked by KUNodes.
√ Nodes for key updates output by KUNodes.
id1 id2 id3 id4 id5 id6 id7 id8
No user is revoked.
id1 id2 id3 id4 id5 id6 id7 id8
User id3 is revoked.

Fig. 2. A pictorial description about how the KUNodes algorithm works.

formal definition of the KUNodes algorithm.

```
KUNodes$(\mathrm{BT}, rl, t)$
  $X, Y \leftarrow \emptyset.$
  $\forall\ (\theta_i, t_i) \in rl,$ if $t_i \leq t,$ then add $\mathrm{Path}(\theta_i)$ to $X$.
  $\forall\ x \in X,$ if $x_l \notin X,$ then add $x_l$ to $Y;$ if $x_r \notin X,$ then add $x_r$ to $Y$.
  If $Y = \emptyset,$ then add **root** to $Y$.
  Return $Y$.
```

3 Framework and Security Model

In this section, we describe the framework and security definition of SR-ABE.

3.1 Framework

An SR-ABE scheme involves four types of entities: a key generation center (KGC), data owners, data users and a untrusted server, and consists of nine algorithms given below. We assume that the server keeps a list of tuples (identity, attribute set, public attribute-key), i.e., (id, $\mathbf{A}$, $pk_{id}^{\mathbf{A}}$).

- Setup(1^{λ}) $\rightarrow$ (par, msk, rl, st). Taking a security parameter λ as the input, this algorithm outputs the public parameter par, the master private key msk, an initially empty revocation list rl and a state st. This algorithm is run by the KGC.
- UserKG(par, id) $\rightarrow$ (sk_{id}, pk_{id}). Taking the public parameter par and an identity as the input, this algorithm outputs a public and private user-key pair (sk_{id}, pk_{id}). This algorithm is run by each data user.
- PubKG(par, msk, id, pk_{id}, $\mathbf{A}$, st) $\rightarrow$ ($pk_{id}^{\mathbf{A}}$, st). Taking the public parameter par, the master private key msk, an identity id with a public user-key pk_{id} and a set of attributes $\mathbf{A}$, and a state st as the input, this algorithm outputs a public attribute-key $pk_{id}^{\mathbf{A}}$ for user id possessing an attribute set $\mathbf{A}$ and an updated state st. This algorithm is run by the KGC, and ($pk_{id}^{\mathbf{A}}$, st) is sent to the untrusted server.
- TKeyUp(par, msk, t, rl, st) $\rightarrow$ (tku_t, st). Taking the public parameter par, the master private key msk, a time period t, a revocation list rl and a state st as the input, this algorithm outputs a key update message tku_t and an updated state st. This algorithm is run by the KGC, and (tku_t, st) is sent to the server.
- TranKG(par, id, $pk_{id}^{\mathbf{A}}$, tku_t) $\rightarrow$ $tk_{id,t}$. Taking the public parameter par, an identity id with the corresponding public attribute-key $pk_{id}^{\mathbf{A}}$ and a key update message tku_t as the input, this algorithm outputs a transformation key $tk_{id,t}$ for user id in time period t. This algorithm is run by the server.
- Encrypt(par, ($\mathbb{M}$, ρ), t, M) $\rightarrow$ CT. Taking the public parameter par, an access structure ($\mathbb{M}$, ρ), a time period t and a message M as the input, this algorithm outputs a ciphertext CT. This algorithm is run by each data owner, and CT will be stored in the cloud.

- Transform(par, id, $\mathbf{A}$, $tk_{id,t}$, CT) → CT$'$/⊥. Taking the public parameter par, an identity id with the corresponding transformation key $tk_{id,t}$ and a ciphertext CT as the input, this algorithm outputs either a partially decrypted ciphertext CT$'$ when the attributes $\mathbf{A}$ associated with the transformation key $tk_{id,t}$ satisfies the access structure of the ciphertext CT or ⊥ indicating the failure of the transformation. This algorithm is run by the server. After the partial decryption, CT$'$ is sent to the data user id.
- Decrypt(par, id, sk_{id}, CT$'$) → M/⊥. Taking the public parameter par, an identity id with a private user-key sk_{id} and a transformed ciphertext CT$'$ as the input, this algorithm outputs a message M or a failure symbol ⊥. This algorithm is run by a data user id.
- Revoke(id, t, rl, st) → rl. Taking an identity id to be revoked, a time period t, a revocation list rl and a state st, this algorithm outputs an updated revocation list rl. This algorithm is run by the KGC.

The correctness of an SR-ABE scheme requires that for any security parameter λ and any message M, if the data user id is not revoked at time period t, and if all parties follow the described algorithms as above, we have Decrypt(par, sk_{id}, CT$'$) = M.

3.2 Security Model

Below we describe the security definition of indistinguishability under chosen plaintext attacks (IND-CPA security) for SR-ABE between an adversary algorithm $\mathcal{A}$ and a challenger algorithm $\mathcal{B}$.

- Setup. Algorithm $\mathcal{B}$ runs the setup algorithm, and gives the public parameter par to algorithm $\mathcal{A}$, and keeps the master private key msk, an initially empty revocation list rl and a state st.
- Phase 1. Algorithm $\mathcal{A}$ adaptively issues a sequence of following queries to algorithm $\mathcal{B}$.
  - Private-User-Key oracle. Algorithm $\mathcal{A}$ issues a private user-key query on an identity id. Algorithm $\mathcal{B}$ returns sk_{id} by running UserKG(par, id).
    Note that once algorithm $\mathcal{B}$ runs UserKG(par, id), it adds (id, pk_{id}, sk_{id}) to a list so that the same (sk_{id}, pk_{id}) is used for all queries on id.
  - Public-Attribute-Key oracle. Algorithm $\mathcal{A}$ issues a public attribute-key query on an identity id and an attribute set $\mathbf{A}$. Algorithm $\mathcal{B}$ returns $pk_{id}^{\mathbf{A}}$ by running UserKG(par, id) (if id has not been issued to the Private-User-Key oracle), PubKG(par, msk, id, pk_{id}, $\mathbf{A}$, st).
  - Transformation-Key-Update oracle. Algorithm $\mathcal{A}$ issues a key update query on a time period t. Algorithm $\mathcal{B}$ runs TKeyUp(par, msk, t, rl, st) and returns tku_t.
  - Transformation-Key oracle. Algorithm $\mathcal{A}$ issues a transformation key query on a time period t and an identity id with an attribute set $\mathbf{A}$. Algorithm $\mathcal{B}$ returns $tk_{id,t}$ by running UserKG(par, id) (if id has not been issued to the Private-User-Key oracle), PubKG(par, msk, id, pk_{id}, $\mathbf{A}$, st), TKeyUp(par,

msk, t, rl, st), TranKG(par, id, $pk_{id}^{\mathbf{A}}$, tku_t). Note that this oracle cannot be queried on a time period t before a transformation key update oracle has been queried on t.

- Revocation oracle. Algorithm $\mathcal{A}$ issues a revocation query on an identity id and a time period t. Algorithm $\mathcal{B}$ runs Revoke(id, t, rl, st) and outputs an updated revocation list rl.Note that a time period t on which a transformation key update query has been issued cannot be issued to this oracle.

- Challenge. Algorithm $\mathcal{A}$ outputs two messages M_0^*, M_1^* of the same size, an access structure ($\mathbb{M}^*$, ρ^*) and a time period t^* satisfying the following constraints.
  1. Case 1: if (1) an identity id^* has been queried to the Private-User-Key oracle, and (2) ($\mathbb{M}^*$, ρ^*) can be satisfied by a query on (id^*, $\mathbf{A}^*$) issued to the Public-Attribute-Key oracle, then (1) the revocation oracle must be queried on (id^*, t) on $t = t^*$ or any t occurs before t^*, and (2) the Transformation-Key oracle cannot be queried on (id^*, t^*).
  2. Case 2: if an identity id^* whose attribute set $\mathbf{A}^*$ can be satisfied by the challenge access structure ($\mathbb{M}^*$, ρ^*) is not revoked at or before t^*, then id^* should not be previously queried to the Private-User-Key oracle.

  Algorithm $\mathcal{B}$ randomly chooses $\gamma \in \{0, 1\}$, and forwards the challenge ciphertext CT* to algorithm $\mathcal{A}$ by running Encrypt(par, ($\mathbb{M}^*$, ρ^*), t^*, M_γ^*).
- Phase 2. Algorithm $\mathcal{A}$ continues issuing queries to algorithm $\mathcal{B}$ as in Phase 1, following the restrictions defined in the Challenge phase.
- Guess. Algorithm $\mathcal{A}$ makes a guess γ' for γ, and it wins the game if $\gamma' = \gamma$.

The advantage of algorithm $\mathcal{A}$ in this game is defined as $\Pr[\gamma = \gamma'] - 1/2$. An SR-ABE scheme is IND-CPA secure if any probabilistic polynomial time (PPT) adversary has at most a negligible advantage in the security parameter λ. In addition, an SR-ABE scheme is said to be selectively IND-CPA secure if an Init stage is added before the Setup phase where algorithm $\mathcal{A}$ commits to the challenge access structure ($\mathbb{M}^*$, ρ^*) (and the challenge time period t^*) which it attempts to attack.

Remark. Seo and Emura [23] defined a security model to prevent a realistic threat called decryption key exposure attacks such that no information of the plaintext is revealed from a ciphertext even if all (short-term) decryption keys of a "different time period" are exposed, which the revocable ABE schemes in [1,5,9,21] following the Boldyreva-Goyal-Kumar technique cannot resist[5]. To cover such attacks in our IND-CPA security model, different from those previous security notions [1,5,9,21] in revocable ABE, the adversary in our CP-ABE definition is given access to an additional Transformation-Key oracle, since the decryption key generated by a data user in a normal ABE scheme is now created by the server and renamed as transformation key in our SR-ABE scheme.

[5] This does not contradict with the security proofs of these schemes, because such attacks are excluded from their security models.

4 Server-Aided Revocable Attribute-Based Encryption

In this section, we present a construction of SR-ABE, and analyze its security.

4.1 Construction

Assume that both the attribute space and the time space are Z_p, and the message space is G_1. The proposed SR-ABE scheme, which is based on the CP-ABE scheme in [20], consists of the following algorithms.

- Setup. This algorithm takes a security parameter λ as the input. It randomly chooses a group G of prime order p with $g \in G$ being the corresponding generator, and defines a bilinear map $\hat{e} : G \times G \to G_1$. Additionally, it randomly chooses $u, h, u_0, h_0, w, v \in G$, $\alpha \in Z_p$. Let rl be an empty list storing revoked users and BT be a binary tree with at least N leaf nodes. Define two functions F_1 and F_2 to map any element y in Z_p to an element in G by $F_1(y) = u^y h$ and $F_2(y) = {u_0}^y h_0$. The public parameter is $par = (g, w, v, u, h, u_0, h_0, \hat{e}(g, g)^\alpha)$ along with rl and st, where st is a state which is set to be BT. The master private key is $msk = \alpha$.
- UserKG. This algorithm takes the public parameter par and an identity id as the input. It randomly chooses $\beta \in Z_p$, and outputs a private and public user-key pair $(sk_{id}, pk_{id}) = (\beta_{id}, g^{\beta_{id}})$ for user id.
- PubKG. This algorithm takes the public parameter par, the master private key msk, an identity id with a public key pk_{id} and an attribute set $\mathbf{A}$, and a state st as the input. Let A_1, ..., A_k be the elements of $\mathbf{A}$. It firstly chooses an undefined leaf node θ from the binary tree BT, and stores id in this node. Then, for each node $x \in \text{Path}(\theta)$, it runs as follows.
  1. It fetches g_x from the node x. If x has not been defined, it randomly chooses $g_x \in G$, computes $g'_x = {pk_{id}}^\alpha / g_x$, and stores g_x in the node x.
  2. It randomly chooses $r_x, r_{x,1}, ..., r_{x,k} \in Z_p$, and computes

$$P_{x,1} = g'_x \cdot w^{r_x}, \quad P_{x,2} = g^{r_x}, \quad P^{(i)}_{x,3} = g^{r_{x,i}}, \quad P^{(i)}_{x,4} = F_1(A_i)^{r_{x,i}} \cdot v^{-r_x}.$$

  3. It outputs $pk^{\mathbf{A}}_{id} = \{x, P_{x,1}, P_{x,2}, P^{(i)}_{x,3}, P^{(i)}_{x,4}\}_{x \in \text{Path}(\theta), i \in [1,k]}$ as the public attribute-key and an updated state st.
- TKeyUp. This algorithm takes the public parameter par, the master private key msk, a time period t, a revocation list rl and a state st as the input. For all $x \in$ KUNodes(BT, rl, t), it fetches g_x (note that g_x is always predefined in the PubKG algorithm) from the node x. It then randomly chooses $s_x \in Z_p$, and outputs the transformation key update information $tku_t = \{x, Q_{x,1}, Q_{x,2}\}_{x \in \text{KUNodes(BT, } rl, \, t)}$ where $Q_{x,1} = g_x \cdot F_2(t)^{s_x}$, $Q_{x,2} = g^{s_x}$.
- TranKG. This algorithm takes the public parameter par, an identity id with a public attribute-key $pk^{\mathbf{A}}_{id}$ and the transformation key update information tku_t as the input. Denote I as Path(θ), J as KUNodes(BT, rl, t). It parses $pk^{\mathbf{A}}_{id}$ as $\{x, P_{x,1}, P_{x,2}, P^{(i)}_{x,3}, P^{(i)}_{x,4}\}_{x \in I, i \in [1,k]}$, tku_t as $\{x, Q_{x,1}, Q_{x,2}\}_{x \in J}$ for some set of

nodes I, J. If $I \cap J = \emptyset$, it returns $\perp$. Otherwise, for any node $x \in I \cap J$, it randomly chooses r'_x, $r'_{x,1}$, ..., $r'_{x,k}$, $s'_x \in Z_p$, and computes

$$\begin{aligned} tk_1 &= P_{x,1} \cdot Q_{x,1} \cdot w^{r'_x} \cdot F_2(t)^{s'_x} = pk_{id}{}^{\alpha} \cdot w^{r_x + r'_x} \cdot F_2(t)^{s_x + s'_x}, \\ tk_2 &= P_{x,2} \cdot g^{r'_x} = g^{r_x + r'_x}, \quad tk_3^{(i)} = P_{x,3}^{(i)} \cdot g^{r'_{x,i}} = g^{r_{x,i} + r'_{x,i}}, \\ tk_4^{(i)} &= P_{x,4}^{(i)} \cdot F_1(A_i)^{r'_{x,i}} \cdot v^{-r'_x} = F_1(A_i)^{r_{x,i} + r'_{x,i}} \cdot v^{-(r_x + r'_x)}, \\ tk_5 &= Q_{x,2} \cdot g^{s'_x} = g^{s_x + s'_x}. \end{aligned}$$

It outputs the transformation key $tk_{id,t} = (tk_1, tk_2, \{tk_3^{(i)}, tk_4^{(i)}\}_{i \in [1,k]}, tk_5)$.

- Encrypt. This algorithm takes the public parameter par, an LSSS access structure $(\mathbb{M}, \rho)$, a time period t and a message M as the input. Let $\mathbb{M}$ be a $l \times n$ matrix. It randomly chooses a vector $\overrightarrow{v} = (\mu, y_2, ..., y_n)^{\perp} \in Z_p^n$. These values will be used to share the encryption exponent μ. For $i = 1$ to l, it calculates $v_i = \mathbb{M}_i \cdot \overrightarrow{v}$ where $\mathbb{M}_i$ is the i-th row of $\mathbb{M}$. In addition, it randomly chooses μ_1, ..., $\mu_l \in Z_p$, and outputs the ciphertext CT $= ((\mathbb{M}, \rho), t, C_0, C_1, \{C_2^{(i)}, C_3^{(i)}, C_4^{(i)}\}_{i \in [1,l]}, C_5)$ where

$$\begin{aligned} C_0 = \hat{e}(g,g)^{\alpha\mu} \cdot M, \quad C_1 = g^{\mu}, \quad C_2^{(i)} = w^{v_i} \cdot v^{\mu_i}, \\ C_3^{(i)} = F_1(A_i)^{-\mu_i}, \quad C_4^{(i)} = g^{\mu_i}, \quad C_5 = F_2(t)^{\mu}. \end{aligned}$$

- Transform. This algorithm takes the public parameter par, an identity id with a transformation key $tk_{id,t}$ over an attribute set $\mathbf{A}$ and a time period t and a ciphertext CT over an access structure $(\mathbb{M}, \rho)$ and the same time period t as the input. Suppose that $\mathbf{A}$ satisfies the access structure $(\mathbb{M}, \rho)$. Let I be defined as $I = \{i : \rho(i) \in \mathbf{A}\}$. Denote by $\{w_i \in Z_p\}_{i \in I}$ a set of constants such that if $\{v_i\}$ are valid shares of any secret μ according to $\mathbb{M}$, then $\sum_{i \in I} w_i v_i = \mu$. It parses CT, and outputs the transformed ciphertext CT$' = (C'_0, C_0)$ where

$$C'_0 = \frac{\prod_{i \in I} \left(\hat{e}(C_2^{(i)}, tk_2)\hat{e}(C_3^{(i)}, tk_3^{(i)})\hat{e}(C_4^{(i)}, tk_4^{(i)})\right)^{w_i} \hat{e}(C_5, tk_5)}{\hat{e}(C_1, tk_1)} = \frac{1}{\hat{e}(g, pk_{id}{}^{\alpha})^{\mu}}.$$

- Decrypt. This algorithm takes the public parameter par, an identity id with a private user-key sk_{id} and a transformed ciphertext CT$'$ as the input. It outputs the message M as $M = (C'_0)^{1/\beta} \cdot C_0$.
- Revoke. This algorithm takes an identity id, a time period t, a revocation list rl and a state st as the input. For all the nodes x associated with identity id, it adds (x, t) to rl, and outputs the updated rl.

Notes and Comments. In the above scheme, g'_x in the PubKG algorithm can also be set as $g^{\alpha+\beta_{id}}/g_x$ such that the KGC runs the UserKG algorithm as follows. For each id, the KGC randomly chooses β_{id}, r, r_1, ..., $r_k \in Z_p$, and outputs a private user-key $sk_{id} = \{K_1, K_2, K_3^{(i)}, K_4^{(i)}\}_{i \in [1,k]}$, where

$$K_1 = g^{\beta_{id}} \cdot w^r, \quad K_2 = g^r, \quad K_3^{(i)} = g^{r_i}, \quad K_4^{(i)} = F_1(A_i)^{r_i} \cdot v^{-r}.$$

However, this requires a secure channel between the KGC and each data user for key transmission. In addition, the KGC possesses all secrets of data users. Lastly, since this key structure follows that in the basic Rouselakis-Waters CP-ABE scheme [20], each data user's computational cost in decryption could not be mitigated, and their storage sizes of private keys are linear to the numbers of the attributes entitled to them.

Remark. Note that the techniques applied in our SR-ABE construction can be used to realize other cryptographic primitives.

- Server-aided revocable KP-ABE. Since our SR-ABE construction uses the same binary tree data structure as in the revocable KP-ABE scheme [1], it is not difficult to see that the technique of having an untrusted server to facilitate computation used in our construction can be applied in a straightforward manner to realize server-aided revocable KP-ABE.
- Server-aided revocable IBE with efficient decryption. In our SR-ABE scheme, we embed a public user-key into the attribute-key such that the server is only able to partially decrypt a ciphertext, and leaves the partially decrypted ciphertext to user for fully decryption using her private user-key. Such a gadget can be easily adopted in the SR-IBE scheme in [19] to reduce data users' decryption costs and remove secure channels for key distribution.

4.2 Security

Theorem 1. *Under the decisional* $(q-1)$ *problem, our SR-ABE scheme is selectively IND-CPA secure.*

Proof. The proof is divided into two cases. In Case 1, it is assumed that an identity id^* whose attribute set $\mathbf{A}^*$ satisfying the challenge access structure $(\mathbb{M}^*, \rho^*)$ is revoked at or before the challenge time period t^*. In Case 2, it is assumed that an identity id^* whose attribute set $\mathbf{A}^*$ satisfying the challenge access structure $(\mathbb{M}^*, \rho^*)$ is not revoked at or before the challenge time period t^*. Briefly speaking, the adversary is allowed to issue a private user-key query on id^* in Case 1, while this query is prohibited in Case 2. We detail the proof in the full version of this paper[6]. □

4.3 Comparison

To our knowledge, in addition to our work in this paper, [1,5,21,26] are also revocable ABE schemes from bilinear pairings (excluding dual vector pairing spaces [21]) in the prime-order groups. Recall that our goal in this paper is to achieve indirect user revocation in a CP-ABE system by delegating data users' workloads to an untrusted server such that the KGC indirectly accomplishes user revocation by stopping updating the keys for revoked data users. In [5], a KP-ABE scheme with indirect revocation is proposed where the KGC enables user

[6] Please contact the authors for it.

Table 1. Comparison between our SR-ABE scheme and existing revocable ABE (R-ABE) schemes from standard bilinear pairings in the prime-order groups.

	R-ABE in [5]	R-ABE in [1]	R-ABE in [26]	R-ABE in [21]	Our SR-ABE
Revocation Mode	Indirect	Indirect & Direct	Direct	Indirect	Indirect
Type of ABE	KP-ABE	KP-ABE	CP-ABE	KP-ABE & CP-ABE	CP-ABE
Server	–	–	Semi-trust	–	Untrust
Key Exposure Resistance	No	No	–	No	Yes
Security	Selective	Selective	Selective	Selective	Selective
Secure Channel	Yes	Yes	Yes	Yes	No
Size of Key Updates	$O(R\log(\frac{N}{R}))$	$O(R\log(\frac{N}{R}))$	–	$O(R\log(\frac{N}{R}))$	$O(R\log(\frac{N}{R}))$
Size of Key Stored by Data User	$O(l\log N)$	$O(l\log N)$	$O(1)$	$O(l\log N)$ & $O(k\log N)$	$O(1)$
Computation Cost in Decrypt	$\geq 2(\mathrm{E}+\mathrm{P})$	$\geq 3\mathrm{E}+4\mathrm{P}$	E	$\geq \mathrm{E}+\mathrm{P}$	E

revocation by stopping posting key update information for revoked data users, thereby forcing revoked data users to be unable to update their decryption keys. A hybrid revocable KP-ABE system is given in [1], which allows a data owner to select either direct or indirect revocation mode when encrypting a message. In [26], a revocable ABE scheme is put forth by giving the direct revocation capability to a semi-trusted server, where the server shares part of the decryption capability of the data users and stops the decryption operation for any revoked data users. A generic way to realize ABE supporting dynamic credentials is provided in [21], where the KGC indirectly accomplishes revocation by stopping updating the keys for revoked data users.

Table 1 compares our SR-ABE scheme with revocable ABE schemes under prime-order groups in [1,5,21,26]. Let N be the number of all data users, R be the number of revoked data users, l be the number of attributes presented in an access structure, and k be the size of the attribute set associated with an attribute-key. Also, let "–" denote not-applicable, "E" denote exponentiation operation, and "P" denote pairing operation, respectively. It is straightforward to see from Table 1 that the schemes in [1,5,21] require secure channels between the KGC and every data user for key transmission, and every data user to keep a private key of which the size is determined by their attributes and the associated nodes in the predefined binary tree. While the scheme in [26] does not require every data user to store a key of large size but requires a secure channel between the KGC and the semi-trusted server, and is subject to collusion attacks between the semi-trusted server and revoked data users. Clearly, our SR-ABE scheme has an edge over previous solutions in that it does not require any secure channels between the system participants, and is secure against collusion attacks between the untrusted server and revoked data users. Also, our SR-ABE scheme achieves desirable efficiency in decryption run by data users, which only requires one exponentiation operation.

5 Conclusions

In this paper, we introduced a notion called server-aided revocable attribute-based encryption (SR-ABE) to achieve efficient user revocation in attribute-based encryption (ABE). We formally defined the (selective) IND-CPA security for SR-ABE, proposed a concrete construction of SR-ABE in terms of ciphertext-policy attribute-based encryption (CP-ABE), and then proved that the proposed SR-ABE scheme is selectively IND-CPA secure. Compared with the previous revocable ABE schemes, our SR-ABE scheme has three salient advantages. First, our SR-ABE scheme delegates almost all computational overheads of data users resulted in key updates to an untrusted server. Second, instead of storing a private key, of which the size is logarithmic to the number of data users, by each data user as in most of the existing revocable ABE schemes, each data user in our SR-ABE scheme only needs to keep a private key of one group element. Third, in our SR-ABE scheme, most of the computational cost in decryption is delegated to the untrusted server, and a data user is only required to perform one exponentiation operation to decrypt a ciphertext. Besides constructing server-aided revocable CP-ABE schemes, the same techniques introduced in this paper can be easily applied to build server-aided revocable key-policy ABE schemes and IBE schemes.

Acknowledgments. This research work is supported by the Singapore National Research Foundation under the NCR Award No. NRF2014NCR-NCR001-012, the National Natural Science Foundation of China under the Grant No. 61502400 and the Foundation of Sichuan Educational Committee under the Grant No. 16ZB0140.

References

1. Attrapadung, N., Imai, H.: Attribute-based encryption supporting direct/indirect revocation modes. In: Parker, M.G. (ed.) Cryptography and Coding 2009. LNCS, vol. 5921, pp. 278–300. Springer, Heidelberg (2009)
2. Attrapadung, N., Imai, H.: Conjunctive broadcast and attribute-based encryption. In: Shacham, H., Waters, B. (eds.) Pairing 2009. LNCS, vol. 5671, pp. 248–265. Springer, Heidelberg (2009)
3. Baek, J., Zheng, Y.: Identity-based threshold decryption. In: Bao, F., Deng, R., Zhou, J. (eds.) PKC 2004. LNCS, vol. 2947, pp. 262–276. Springer, Heidelberg (2004)

4. Beimel, A.: Secure schemes for secret sharing and key distribution. Ph.D. thesis, Israel Institute of Technology, June 1996
5. Boldyreva, A., Goyal, V., Kumar, V.: Identity-based encryption with efficient revocation. In: Proceedings of the ACM Conference on Computer and Communications Security, CCS 2008, Alexandria, Virginia, USA, 27–31 October 2008, pp. 417–426. ACM (2008)
6. Boneh, D., Boyen, X.: Efficient selective identity-based encryption without random oracles. J. Cryptology **24**(4), 659–693 (2011)
7. Boneh, D., Ding, X., Tsudik, G., Wong, C.: A method for fast revocation of publickey certificates and security capabilities. In: 10th USENIX Security Symposium, 13–17 August 2001, Washington, D.C., USA. USENIX (2001)
8. Boneh, D., Franklin, M.: Identity-based encryption from the weil pairing. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, p. 213. Springer, Heidelberg (2001)
9. Cui, H., Deng, R.H.: Revocable and decentralized attribute-based encryption. Comput. J. doi:10.1093/comjnl/bxw007
10. Ding, X., Tsudik, G.: Simple identity-based cryptography with mediated RSA. In: Joye, M. (ed.) CT-RSA 2003. LNCS, vol. 2612, pp. 193–210. Springer, Heidelberg (2003)
11. Hanaoka, Y., Hanaoka, G., Shikata, J., Imai, H.: Identity-based hierarchical strongly key-insulated encryption and its application. In: Roy, B. (ed.) ASIACRYPT 2005. LNCS, vol. 3788, pp. 495–514. Springer, Heidelberg (2005)
12. Horváth, M.: Attribute-based encryption optimized for cloud computing. In: Italiano, G.F., Margaria-Steffen, T., Pokorný, J., Quisquater, J.-J., Wattenhofer, R. (eds.) SOFSEM 2015-Testing. LNCS, vol. 8939, pp. 566–577. Springer, Heidelberg (2015)
13. Lewko, A., Waters, B.: Decentralizing attribute-based encryption. In: Paterson, K.G. (ed.) EUROCRYPT 2011. LNCS, vol. 6632, pp. 568–588. Springer, Heidelberg (2011)
14. Li, J., Li, J., Chen, X., Jia, C., Lou, W.: Identity-based encryption with outsourced revocation in cloud computing. IEEE Trans. Comput. **64**(2), 425–437 (2015)
15. Li, Q., Xiong, H., Zhang, F.: Broadcast revocation scheme in composite-order bilinear group and its application to attribute-based encryption. IJSN **8**(1), 1–12 (2013)
16. Liang, K., Liu, J.K., Wong, D.S., Susilo, W.: An efficient cloud-based revocable identity-based proxy re-encryption scheme for public clouds data sharing. In: Kutyłowski, M., Vaidya, J. (eds.) ICAIS 2014, Part I. LNCS, vol. 8712, pp. 257–272. Springer, Heidelberg (2014)
17. Libert, B., Quisquater, J.: Efficient revocation and threshold pairing based cryptosystems. In: Proceedings of the Twenty-Second ACM Symposium on Principles of Distributed Computing, PODC 2003, Boston, Massachusetts, USA, 13–16 July 2003, pp. 163–171. ACM (2003)
18. Naor, D., Naor, M., Lotspiech, J.: Revocation and tracing schemes for stateless receivers. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, p. 41. Springer, Heidelberg (2001)
19. Qin, B., Deng, R.H., Li, Y., Liu, S.: Server-aided revocable identity-based encryption. In: Pernul, G., Y A Ryan, P., Weippl, E. (eds.) ESORICS. LNCS, vol. 9326, pp. 286–304. Springer, Heidelberg (2015). doi:10.1007/978-3-319-24174-6_15
20. Rouselakis, Y., Waters, B.: Practical constructions and new proof methods for large universe attribute-based encryption. In: ACM SIGSAC Conference on Computer and Communications Security, CCS 2013, Berlin, Germany, 4–8 November 2013, pp. 463–474. ACM (2013)

21. Sahai, A., Seyalioglu, H., Waters, B.: Dynamic credentials and ciphertext delegation for attribute-based encryption. In: Safavi-Naini, R., Canetti, R. (eds.) CRYPTO 2012. LNCS, vol. 7417, pp. 199–217. Springer, Heidelberg (2012)
22. Sahai, A., Waters, B.: Fuzzy identity-based encryption. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 457–473. Springer, Heidelberg (2005)
23. Seo, J.H., Emura, K.: Revocable identity-based encryption revisited: security model and construction. In: Kurosawa, K., Hanaoka, G. (eds.) PKC 2013. LNCS, vol. 7778, pp. 216–234. Springer, Heidelberg (2013)
24. Wan, Z., Liu, J., Deng, R.H.: HASBE: a hierarchical attribute-based solution for flexible and scalable access control in cloud computing. IEEE Trans. Inf. Forensics Secur. **7**(2), 743–754 (2012)
25. Waters, B.: Ciphertext-policy attribute-based encryption: an expressive, efficient, and provably secure realization. In: Catalano, D., Fazio, N., Gennaro, R., Nicolosi, A. (eds.) PKC 2011. LNCS, vol. 6571, pp. 53–70. Springer, Heidelberg (2011)
26. Yang, Y., Ding, X., Lu, H., Wan, Z., Zhou, J.: Achieving revocable fine-grained cryptographic access control over cloud data. In: Desmedt, Y. (ed.) ISC 2013. LNCS, vol. 7807, pp. 293–308. Springer, Heidelberg (2015). doi:10.1007/978-3-319-27659-5_21