10-2019

Generic construction of ElGamal-type attribute-based encryption schemes with revocability and dual-policy

Shengmin XU
Singapore Management University, smxu@smu.edu.sg

Yinghui ZHANG

Yingjiu LI

Ximeng LIU

Guomin YANG
Singapore Management University

Citation

XU, Shengmin; ZHANG, Yinghui; LI, Yingjiu; LIU, Ximeng; and YANG, Guomin. Generic construction of ElGamal-type attribute-based encryption schemes with revocability and dual-policy. (2019). *Proceedings of the 15th EAI International Conference, SecureComm 2019, Orlando, FL, USA, October 23–25*. 184-204.
Available at: https://ink.library.smu.edu.sg/sis_research/5190

Generic Construction of ElGamal-Type Attribute-Based Encryption Schemes with Revocability and Dual-Policy

Shengmin Xu[1], Yinghui Zhang[1,2(✉)], Yingjiu Li[2], Ximeng Liu[3,4], and Guomin Yang[5]

[1] National Engineering Laboratory for Wireless Security, Xi'an University of Posts and Telecommunications, Xi'an 710121, China
yhzhaang@163.com
[2] School of Information Systems, Singapore Management University, Singapore, Singapore
[3] College of Mathematics and Computer Science, Fuzhou University, Fuzhou, China
[4] Key Lab of Information Security of Network Systems, Fuzhou University, Fuzhou, Fujian, China
[5] School of Computing and Information Technology, University of Wollongong, Wollongong, Australia

Abstract. Cloud is a computing paradigm for allowing data owners to outsource their data to enjoy on-demand services and mitigate the burden of local data storage. However, secure sharing of data via cloud remains an essential issue since the cloud service provider is untrusted. Fortunately, asymmetric-key encryption, such as identity-based encryption (IBE) and attribute-based encryption (ABE), provides a promising tool to offer data confidentiality and has been widely applied in cloud-based applications. In this paper, we summarize the common properties of most of IBE and ABE and introduce a cryptographic primitive called ElGamal type cryptosystem. This primitive can be used to derive a variety of ABE schemes. To illustrate the feasibility, we present generic constructions of revocable attribute-based encryption and dual-policy attribute-based encryption with formal definitions and security proofs. By applying our proposed generic constructions, we also present instantiations of these schemes. Furthermore, we demonstrate the high performance of the proposed schemes via experiments.

Keywords: ElGamal-type cryptosystem · Attribute-based encryption

1 Introduction

Public-key encryption is the fundamental primitive of public-key cryptography, which removes the key-agreement process in traditional symmetric-key encryption to facilitate data sharing via the certificate list. However, conventional public-key infrastructure is vulnerable to certificate management. To address this issue, identity-based encryption (IBE) [10] was proposed to provide a new

S. Chen et al. (Eds.): SecureComm 2019, LNICST 305, pp. 184–204, 2019.
https://doi.org/10.1007/978-3-030-37231-6_10

paradigm by utilizing the user identity rather than searching the certificate of the receiver. Unfortunately, IBE only provides coarse-level data sharing. To overcome this drawback, attribute-based encryption (ABE) [26] was introduced. There are mainly two types of standard ABE systems: key-policy ABE (KP-ABE) and ciphertext-based ABE (CP-ABE) and they are useful in different applications. KP-ABE provides the content-based access control by specifying the receiver's policy over ciphertext's attributes for managing the accessing of sensitive information. CP-ABE offers the role-based access control by specifying the ciphertext's policy over the receiver's policy for controlling the data receiver.

IBE. Boneh and Franklin [10] proposed the first practical IBE by transforming ElGamal encryption [12] in finite fields to bilinear groups. To improve the performance, Boneh and Boyen [8,9] proposed the selectively secure IBE without random oracles and Waters [28] introduced an adaptively secure IBE in the standard model. Gentry [13] proposed an adaptive security IBE with the constant-size parameter based on the interactive assumption. Abdalla et al. [1] then proposed IBE with wildcard operation. Lewko and Waters [18] design the first adaptive security IBE with the standard assumption by applying dual system encryption [29], which also used to design the adaptive security ABE [17].

KP-ABE. In KP-ABE, the key generation center (KGC) generates the users' secret keys based on corresponding access trees, and ciphertexts are encrypted over a set of attributes. The encryptor has no control over who has access to the data except by choosing descriptive attributes for the data. The initial work was introduced by Sahai and Waters [26]. To enrich the expression, Goyal et al. [14] provided KP-ABE with monotonic span programs and Ostrovsky et al. [22] proposed KP-ABE supporting non-monotonic access structures. Attrapadung et al. [3] then proposed KP-ABE with constant-size ciphertexts.

CP-ABE. In CP-ABE, access trees are used to encrypt data and users' secret keys associate a set of attributes. The encryptor has to manage the access tree to specify the users' access right. The seminal work was introduced by Bethencourt et al. [6] with two-level random masking methodology. Waters [30] introduced the first selectively secure CP-ABE under the non-standard assumption, and Rouselakis and Waters [24] provided CP-ABE with the large universe.

Generic ABE Constructions. Generic constructions of ABE have been well studied before. Sahai et al. [25] proposed the generic ABE with piecewise key generation to derive RABE. Chow [11] provided generic ABE with the properties of key partition and ciphertext partition to build RABE with the multi-authority setting. Not surprisingly, self-updatable ABE [16] also applied a similar strategy. The core technique of them are based on the secret-splitting trick, and we also apply this concept to build generic schemes.

Many cloud-based data sharing applications are built based on IBE and ABE since they facilitate data sharing securely. However, consider usability and functionality, directly applying these scheme in cloud-based applications is insufficient. To address this problem, many cryptosystems with practical properties have been proposed, such as public-key cryptosystems with revocation

(e.g., revocable IBE and ABE, RIBE/RABE for short) and dual-policy ABE (DP-ABE) with content-based and role-based access control simultaneously.

RIBE/RABE. RIBE/RABE (as shown in Fig. 1) is an extension of IBE/ABE by providing an efficient revocation mechanism. The issues of revocation have been pointed out in the corresponding seminal works [10,23] and they suggested extending each attribute with an expiration date, e.g., private keys periodically update by representing an attribute as $att\|t$, where att is the real attribute and t is the current date. However, such an approach incurs the heavy workload and unscalable because a secure channel between KGC and each user needs to be established each time. Boldyreva et al. [7] solved this issue by introducing indirect revocation, which divides the (short-term) decryption key into the long-term secret key and the public key-updating material. With this method, KGC only publishes public key-updating material in each revocation epoch. By applying tree-based structure [21], the size of the key-updating material is logarithmic in the number of system users. However, this work suffers decryption key exposure attacks. The frequently used decryption key could be compromised due to a variety of reasons, such as side-channel and key-leakage attacks. The long-term secret key will be compromised if the short-term decryption key is leaked. Seo and Emura [27] provided a strong model with perfect forward secrecy in the identity-based setting and proposed a secure RIBE under this model.

DP-ABE. To make the most advantages of both KP-ABE and CP-ABE, DP-ABE (as shown in Fig. 2) [2,4] was introduced. It is a conjunctively combined between two types of ABE. Ciphertexts are specified access trees and a set of attributes simultaneously, and the secret keys are also required to specified a set of attributes and access trees. We further category DP-ABE into two types: sequential DP-ABE and parallel DP-ABE. In sequential DP-ABE, receivers will be able to decrypt if who pass both restrictions. Interestingly, the sequential DP-ABE is similar to RABE with indirect revocation. In RABE, two restrictions are long-term secret key and public key-updating material, where long-term secret keys are related to the key-policy or the ciphertext-policy depending on the access policy of RABE, and the public key-updating as the restriction based on the revocation mechanism. In parallel DP-ABE (sometimes called one-policy DP-ABE), receivers only need to satisfy one of two limitations to review messages. It is worth to notice that Attrapadung and Yamada [4] provided the generic construction based on ABE and pairing encodings, our generic constructions are based on the different building block ElGamal type cryptosystem, which also can be used to build RABE schemes.

1.1 Contribution

In this paper, we revisit ElGamal-like schemes [12] in both identity-based [1,5,8–10,18,28] and attribute-based [3,6,14,17,20,22,24,26,30] settings and introduce a new primitive called ElGamal type cryptosystem with formal definition and security model by summarizing their common properties. By applying this primitive, we can easily derive a variety of cryptosystems. To illustrate the feasibility

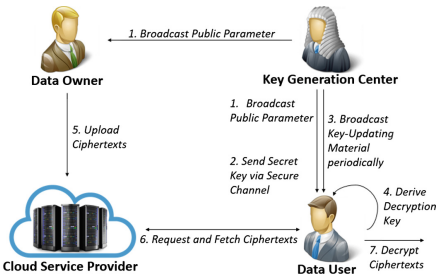

Fig. 1. System model of RABE/RIBE

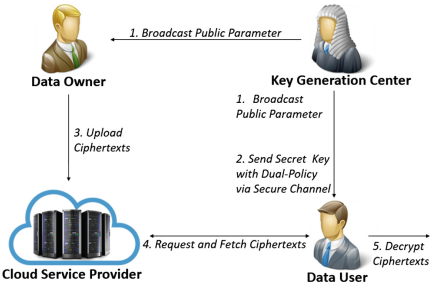

Fig. 2. System model of DP-ABE

of our proposed ElGamal type cryptosystem, we present the generic constructions of RABE with decryption exposure resistance and DP-ABE with parallel and sequential settings.

We first investigate RIBE/RABE and present the generic construction of RABE with decryption key exposure resistance. We should note that our ElGamal type cryptosystem allows the key re-randomization without the master secret key. This re-randomizable property is to remove the relationships among the long-term secret key, the public key-updating material, and the short-term decryption key. Hence, the long-term secret key is secure even both the key-updating material and the decryption key are compromised.

We then investigate DP-ABE schemes and present the concrete schemes of DP-ABE with parallel and sequential settings in the prime-order group. These schemes are the provable security under the proposed models. We then give detailed comparisons and experiment results to demonstrate the usability and high performance of our proposed schemes.

1.2 Outline

In Sect. 2, we introduce some preliminaries including the proposed ElGamal type cryptosystem and its semi-generic construction. In Sect. 3, we present definitions of DP-ABE and RABE. In Sect. 4, we give generic constructions of RABE and DP-ABE and the corresponding formal proofs. The instantiations of these schemes are presented in Sect. 5. In Sect. 6, we provide the analysis of functionality and efficiency. We summarize this paper in Sect. 6.

2 Preliminaries

2.1 Notations

Let $\mathbb{N}$ denote the set of all natural numbers, and for $n \in \mathbb{N}$, we define $[n] := 1, ..., n$. Let $\vec{u} := (u_1, u_2, ..., u_\ell)$ denote a vector of dimension ℓ in $\mathbb{Z}_p$. To simplicity, $X \in \mathcal{X}$ denotes the attributes of key and $Y \in \mathcal{Y}$ represents the attributes of ciphertexts. Depending on the policy in the underlying ABE scheme, X and

Y denote either an attribute set $\mathcal{S}$ or an access policy $\mathbb{A}$, and $\mathcal{X}$ and $\mathcal{Y}$ represent either the attribute universe Ω or the policies $\mathcal{P}$. Let $R : (\mathbb{X}, \mathbb{Y}) \to \{0, 1\}$ denote the result of sufficient condition by inputting key attributes $X \in \mathbb{X}$ and ciphertext attributes $Y \in \mathbb{Y}$, and outputting a bit 0 or 1.

2.2 Bilinear Map

Let $\mathbb{G}$ and $\mathbb{G}_T$ be two cyclic multiplicative groups of prime order p and g be a generator of $\mathbb{G}$. The map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ is said to be an admissible bilinear pairing if the following properties hold true.

- *Bilinearity*: for all $u, v \in \mathbb{G}$ and $a, b \in \mathbb{Z}_p$, $e(u^a, v^b) = e(u, v)^{ab}$.
- *Non-degeneration*: $e(g, g) \neq 1$.
- *Computability*: it is efficient to compute $e(u, v)$ for any $u, v \in \mathbb{G}$.

2.3 Access Structure and Monotone Span Program

Definition 1 (Access Structure [14]). *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 (Monotone Span Program (MSP) [14]). *Let $\mathcal{K}$ be a field and $\{x_1, ..., x_n\}$ be a set of variables. A MSP over $\mathcal{K}$ is labeled matrix $\tilde{M}(\mathbb{M}, \rho)$ where $\mathbb{M}$ is a matrix over $\mathcal{K}$, and ρ is a labeling of the rows of $\mathbb{M}$ by literals from $\{x_1, ..., x_n\}$ (every row is labeled by one literal). A MSP accepts or rejects an input by the following criterion. For every input set $\mathcal{S}$ if literals, define the submatrix $\mathbb{M}_{\mathcal{S}}$ of $\mathbb{M}$ consisting of those rows whose labels are in S, i.e., rows labeled by some i such that $i \in \mathcal{S}$. The MSP $\tilde{M}$ accepts $\mathcal{S}$ if and only if $\vec{1} \in \mathsf{span}(\mathbb{M}_S)$, i.e., some linear combination of the rows of $\mathbb{M}_{\mathcal{S}}$ given the all-one vector $\vec{1}$. The MSP $\tilde{M}$ computes a boolean function f_M if it accepts exactly those input $\mathcal{S}$ where $f_{\mathbb{M}}(\mathcal{S}) = 1$. The size of $\tilde{M}$ is the number of rows in $\mathbb{M}$.*

2.4 Definition of ElGamal Type Cryptosystem

Definition 3 (ElGamal Type Cryptosystem). *ElGamal type cryptosystem $\mathcal{ETC}$ with the key attribute universe $\mathcal{X}$ that supports the ciphertext attribute universe $\mathcal{Y}$ and the message space $\mathcal{M}$ consists of the following five algorithms:*

- $\mathcal{ETC}.\mathsf{Init}(\lambda) \to pp$*: The probabilistic initialization algorithm takes the security parameter $\lambda \in \mathbb{N}$ as input, and outputs the public parameter pp, such as the description of the bilinear group from the bilinear group generator $(g, p, \mathbb{G}, \mathbb{G}_T) \leftarrow \mathcal{G}(\lambda)$.*
- $\mathcal{ETC}.\mathsf{Setup}(pp) \to (pk, msk)$*: The probabilistic setup algorithm takes the parameter pp as input, and outputs the public key pk and the master secret key msk. It is required that the master secret key and the public key are in the form of*

$$msk = (\alpha, ...), \quad pk = (e(g, g)^{\alpha}, ...),$$

where $\alpha \in \mathbb{Z}_p$.

- $\mathcal{ETC}.\mathsf{KeyGen}(msk, X) \rightarrow sk_X$: *The probabilistic key generation algorithm takes the master secret key* msk *and the attributes of the secret key* $X \in \mathcal{X}$ *as input, and outputs the secret key* sk_X. *It is required that the secret key is in the form of* $sk_X = (sk_1, sk_2, sk_3)$ *as:*

$$sk_1 = g^{h(\alpha)} \cdot f(pk, X)^r, \quad sk_2 = g^r,$$

where $r \in \mathbb{Z}_p$, $h(x) \in \mathbb{Z}_p$ *and* $f(x, y) \in \mathbb{G}$. *Note that* sk_3 *is for recording some extra information related attributes of the secret key.*
- $\mathcal{ETC}.\mathsf{Enc}(pk, Y, m) \rightarrow c_Y$: *The probabilistic encryption algorithm takes the public key* pk, *the attributes of the ciphertext* $Y \in \mathcal{Y}$ *and the message* $m \in \mathcal{M}$ *as input, and outputs the ciphertext* c_Y. *It is required that the ciphertext is in the form of* $c_Y = (c_0, c_1, c_2)$ *as:*

$$c_0 = m \cdot e(g, g)^{\alpha s}, \quad c_1 = g^s,$$

where $s \in \mathbb{Z}_p$ *and* c_2 *is some extra information related to attributes of the ciphertext.*
- $\mathcal{ETC}.\mathsf{Dec}(pk, sk_X, c_Y) \rightarrow m$: *The deterministic decryption algorithm takes the public key* pk, *the secret key* sk_X *and the ciphertext* c_Y *as input, and outputs the message* $m \in \mathcal{M}$. *The decryption process is required to be two steps. The first step is to run the sub-decryption algorithm* $\mathcal{D}$ *to get the message hiding component* $e(g, g)^{\alpha s} \leftarrow \mathcal{D}(sk_X, c_1, c_2)$. *The second step is to extract the plaintext by eliminating the message hiding component as* $m = c_0 / e(g, g)^{\alpha s}$.

The consistency condition requires for all $\lambda \in \mathbb{N}$, *all pp output by the initialization algorithm,* pk *and* msk *output by setup algorithm,* $m \in \mathcal{M}$ *and* $R(X, Y) = 1$, *we then have*

$$\mathcal{ETC}.\mathsf{Dec}(pk, sk_X, \mathcal{ETC}.\mathsf{Enc}(pk, Y, m)) = m.$$

Next, we describe the security model called selectively indistinguishable against chosen plaintext attack (sIND-CPA) for ElGamal type cryptosystem.

Definition 4 (sIND-CPA in ElGamal type cryptosystem). *An ElGamal type cryptosystem consists of five algorithms above. For an adversary* $\mathcal{A}$, *we define the following experiment:*

$$\begin{aligned}
&\mathbf{Exp}^{\mathsf{sIND\text{-}CPA}}_{\mathcal{A},\mathcal{ETC}}(\lambda)\\
&\quad Y^* \leftarrow \mathcal{A}(\lambda);\\
&\quad pp \leftarrow \mathcal{ETC}.\mathsf{Init}(\lambda);\\
&\quad (pk, msk) \leftarrow \mathcal{ETC}.\mathsf{Setup}(pp);\\
&\quad (m_0, m_1) \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{KeyGen}}}(pp, pk);\\
&\quad b \leftarrow \{0, 1\};\\
&\quad c^* \leftarrow \mathsf{Enc}(pk, Y^*, m_b);\\
&\quad b' \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{KeyGen}}(\cdot)}(c^*);\\
&\quad \textit{If } b = b' \textit{ return } 1 \textit{ else return } 0.
\end{aligned}$$

$\mathcal{O}_{\mathsf{KeyGen}}(\cdot)$ represents the key generation oracle which allows $\mathcal{A}$ to query on attributes of keys $X \in \mathcal{X}$ except $R(X, Y^*) = 1$ to return the secret key sk_X by running $\mathcal{ETC}.\mathsf{KeyGen}(msk, X)$.
An ElGamal type cryptosystem is said to be sIND-CPA *secure if for any probabilistic polynomial time adversary $\mathcal{A}$, the following advantage is negligible:*

$$\mathbf{Adv}^{\mathsf{sIND\text{-}CPA}}_{\mathcal{A},\mathcal{ETC}}(\lambda) = \left| \Pr[\mathbf{Exp}^{\mathsf{sIND\text{-}CPA}}_{\mathcal{A},\mathcal{ETC}}(\lambda) = 1] - 1/2 \right|$$

2.5 Candidates of ElGamal Type Cryptosystem

The IBE [1,5,8–10,18,28] and ABE [3,6,14,20,22,24,26,30] schemes are the instantiations of ElGamal type cryptosystems. We demonstrate three candidates of them to illustrate the feasibility of our proposed ElGamal type cryptosystem.

IBE. Let r, s denote random numbers over $\mathbb{Z}_p$, $\mathcal{I}$ be the identity space and $\ell = \log_2 \mathcal{I}$ be the length of message space. Waters' IBE [28] is given below.

$$\begin{aligned}
pk &= \left(\underline{e(g,g)^{\alpha}}, u_0, u_1, ..., u_\ell\right), \quad msk := (\underline{\alpha}), \\
sk_{id} &= \left(\underline{g^{\alpha}\left(u_0 \prod_{i\in\mathcal{V}} u_i\right)^{r}}, \underline{g^{r}}\right), \\
c_{id} &= \left(\underline{m \cdot e(g,g)^{\alpha s}}, \underline{g^{s}}, \left(u_0 \prod_{i\in\mathcal{V}} u_i\right)^{s}\right),
\end{aligned}$$

where $\mathcal{V} \in [\ell]$ be the set of all i for which $id[i] = 1$.

KP-ABE. Let $\Delta_{i,\mathsf{J}} = \prod_{j\in\mathsf{J}, j\neq i} \left(\frac{x-j}{i-j}\right)$ denote the Lagrange coefficient for $x, i \in \mathbb{Z}_p$ and $\mathsf{J} \subset \mathbb{Z}$, n denote the maximum size of attributes used in encryption, $\mathbb{M}$ is a matrix over $\mathbb{Z}_p$ with d rows and l columns, ρ is a mapping function that maps any number in the domain $[d]$ to the attribute universe Ω, $\mathcal{S} = (A_1, A_2, ..., A_k)$ is the attribute set and $(\{r_i\}_{i\in[d]}, s)$ represent random numbers in $\mathbb{Z}_p$. Goyal et al's KP-ABE [14] is described as follows.

$$\begin{aligned}
pk &= \left(\underline{e(g,g)^{\alpha}}, \{t_i\}_{i\in[n+1]}\right), \quad msk = (\underline{\alpha}), \\
sk_{\mathbb{A}} &= \left(\{\underline{g^{\mathbb{M}_i \vec{u}_i} T(i)^{r_i}}, \underline{g^{r_i}}\}_{i\in[d]}\right), \\
c_{\mathcal{S}} &= \left(\underline{m \cdot e(g,g)^{\alpha s}}, \underline{g^{s}}, \{T(i)^{s}\}_{\rho(i)\in\mathcal{S}}\right),
\end{aligned}$$

where the vector $\vec{u}$ is a random l dimensional vector over $\mathbb{Z}_p$ *s.t.* $\vec{1} \cdot \vec{u} = \alpha$ and $T(x) = g^{x^n} \prod_{i=1}^{n+1} t_i^{\Delta_{i,[n+1]}(x)}$ be a function to map any index $x \in \mathbb{Z}_p$ to the element in $\mathbb{G}$.

CP-ABE. Let $\{\phi_i\}_{i\in[n]} \in \mathbb{Z}_p$ denote a set of random numbers, Rouselakis and Waters' CP-ABE [24] is given below.

$$\begin{aligned}
pk &= \left(\underline{e(g,g)^{\alpha}}, u, h, w, v\right), \quad msk = (\underline{\alpha}), \\
sk_{\mathcal{S}} &= \left(\underline{g^{\alpha} w^{r}}, \underline{g^{r}}, \{g^{r_i}, (u^{A_i} h)^{r_i} v^{-r}\}_{\rho(i)\in\mathcal{S}}\right), \\
c_{\mathbb{A}} &= \left(\underline{m \cdot e(g,g)^{\alpha s}}, \underline{g^{s}}, \{w^{\mathbb{M}_i \vec{u}_i} v^{\phi_i}, (u^{\rho(i)} h)^{\phi_i}, g^{\phi_i}\}_{i\in[d]}\right),
\end{aligned}$$

where $\vec{u}$ is a l dimensional vector in the domain $\mathbb{Z}_p$ *s.t.* $\vec{u} = (s, u_2, ..., u_l) \in \mathbb{Z}_p^l$ and $r, \{r_i\}_{\rho(i)\in\mathcal{S}}, \{\phi_i\}_{i\in[d]}$ are random numbers over $\mathbb{Z}_p$. Note that referring to Definition 3, we set $h(\alpha) = \alpha$ and $f(pk, X)^r = w^r$ for above CP-ABE setting.

2.6 Tree-Based Revocation Mechanism

Naor et al. [21] introduced a tree-based revocation architecture to reduce the cost of generating and transmitting key updates from linear to logarithmic. Let st be the state representing the tree-based data structure, rl denote the revocation list recording identities of revoked users and the timestamp of revocation and t be the timestamp representing the current revocation epoch. By running subset-cover algorithm $\mathsf{KUNode}(st, rl, t)$, the KGC can derive get the key updates for all non-revoked users with the logarithmic size. When a user wants to join the system, who will be assigned a random identifier $id \in \mathcal{I}$ and an undefined leaf node in st will be labeled this identifier id. The revocation method only requires the user id to store the keys in $\mathsf{Path}(id)$, where $\mathsf{Path}(id)$ denotes nodes from the root to the leaf node id. The details of algorithm $\mathsf{KUNode}(st, rl, t)$ are given in Algorithm 1.

Algorithm 1: Node Selection Algorithm

Input: BT, rl, t
Output: Y
$\mathsf{X}, \mathsf{Y} \leftarrow \emptyset$;
for $(v_i, t_i) \in rl$ **do**
  if $t_i \leq t$ **then**
    $\mathsf{X} \leftarrow \mathsf{X} \cup \mathsf{Path}(v_i)$
for $x \in \mathsf{X}$ **do**
  if $x_l \notin \mathsf{X}$ **then**
    $\mathsf{Y} \leftarrow \mathsf{Y} \cup x_l$
  if $x_r \notin \mathsf{X}$ **then**
    $\mathsf{Y} \leftarrow \mathsf{Y} \cup x_r$
if $\mathsf{Y} = \emptyset$ **then**
  $\mathsf{Y} \leftarrow \mathsf{root}$
return Y.

3 Definition of Revocable ABE and Dual-Policy ABE

In this section, we introduce definitions of RABE and DP-ABE. Specifically, we first introduce the syntax and security model of RABE. Next, we introduce syntaxes of DP-ABE for both parallel and sequential settings and their corresponding security models. We should note that ElGamal type cryptosystem

supports IBE which could be used to manage the timestamp in RABE to efficient revocation rather than ABE to manage the timestamp [25] causing high computation and communication costs. Note that our proposed scheme is easy to apply outsourced ABE [15] since the elegant construction of ElGamal-like encryption to reduce the cost of ciphertext decryption.

3.1 Revocable ABE

Definition 5 (RABE). *An RABE $\mathcal{RABE}$ with key attributes $X, \bar{X}_t \in \mathcal{X} \times \bar{\mathcal{X}}_t$ that support ciphertext $Y, \bar{Y}_t \in \mathcal{Y} \times \bar{\mathcal{Y}}_t$[1], the bounded system lifetime $\mathcal{T}$, an identifier space $\mathcal{I}$, the number of system users $\mathcal{N}$ and the message space $\mathcal{M}$ consists of nine algorithms given below.*

- $\mathcal{RABE}.\mathsf{Setup}(\lambda) \rightarrow (pk, msk, rl, st)$*: The probabilistic setup algorithm takes parameter $\lambda \in \mathbb{N}$ as input, and outputs a public key pk, a master secret key msk, a revocation list rl and a state st.*
- $\mathcal{RABE}.\mathsf{KeyGen}(msk, st, X, id) \rightarrow (sk_{id}, st)$*: The probabilistic key generation algorithm takes the master secret key msk, the state st, the key attributes $X \in \mathcal{X}$ and an identifier $id \in \mathcal{I}$ as input, and outputs the secret key sk_{id} and the state st.*
- $\mathcal{RABE}.\mathsf{KeyUpdate}(msk, st, \bar{X}_t, rl) \rightarrow ku_t$*: The probabilistic key update algorithm takes the master secret key msk, the state st, the key attributes $\bar{X}_t \in \bar{\mathcal{X}}$ associated the timestamp t, the time t and the revocation list rl as input, and outputs the key-updating material ku_t.*
- $\mathcal{RABE}.\mathsf{DKGen}(pk, sk_{id}, ku_t) \rightarrow dk_{id,t}$*: The probabilistic decryption key generation algorithm takes the public key pk, the secret key sk_{id} and key-updating material ku_t as input, and outputs the decryption key $dk_{id,t}$.*
- $\mathcal{RABE}.\mathsf{Enc}(pk, Y, \bar{Y}_t, m) \rightarrow c_{Y,\bar{Y}_t}$*: The probabilistic encryption algorithm takes the public key pk, the ciphertext attribute $Y \in \mathcal{Y}$, the ciphertext attribute $\bar{Y}_t \in \bar{\mathcal{Y}}$ associated with the timestamp $t \in \mathcal{T}$ and a message $m \in \mathcal{M}$ as input, and outputs a ciphertext c.*
- $\mathcal{RABE}.\mathsf{Dec}(pk, dk_{id,t}, c_{Y,\bar{Y}}) \rightarrow m$*: The deterministic decryption algorithm takes the public key pk, the decryption key $dk_{id,t}$ and a ciphertext $c_{Y,\bar{Y}}$ as input, and outputs a message $m \in \mathcal{M}$.*
- $\mathcal{RABE}.\mathsf{Rev}(rl, id, t) \rightarrow rl$*: The deterministic revocation algorithm takes the revocation list rl, an identifier $id \in \mathcal{I}$ and the timestamp $t \in \mathcal{T}$ as input, and outputs the revocation list rl.*

[1] $\bar{X}_t$ and $\bar{Y}_t$ is based on the timestamp t (e.g., the bit representation of the timestamp or the policies derived from its bit representation) which is used to manage user revocation.

3.2 Security Model of RABE

Definition 6 (sIND-CPA **in RABE).** *An RABE consist of seven algorithms in above. For an adversary $\mathcal{A}$, we define the following experiment:*

$$
\begin{aligned}
&\mathbf{Exp}^{\mathsf{sIND\text{-}CPA}}_{\mathcal{A},\mathcal{RABE}}(\lambda)\\
&\quad (Y^*, \bar{Y}_{t^*}) \leftarrow \mathcal{A}(\lambda);\\
&\quad (pk, msk, rl, st) \leftarrow \mathcal{RABE}.\mathsf{Setup}(pp);\\
&\quad (m_0, m_1) \leftarrow \mathcal{A}^{\mathcal{O}}(pp, pk);\\
&\quad b \leftarrow \{0,1\};\\
&\quad c^* \leftarrow \mathsf{Enc}(pk, Y^*, \bar{Y}_{t^*}, m_b);\\
&\quad b' \leftarrow \mathcal{A}^{\mathcal{O}}(c^*);\\
&\quad \textit{If } b = b' \textit{ return } 1 \textit{ else return } 0.
\end{aligned}
$$

$\mathcal{O}$ is a set of oracles, $\{\mathcal{O}_{\mathsf{KeyGen}}(\cdot,\cdot), \mathcal{O}_{\mathsf{KeyUpdate}}(\cdot,\cdot), \mathcal{O}_{\mathsf{Rev}}(\cdot,\cdot), \mathcal{O}_{\mathsf{DKGen}}(\cdot,\cdot)\}$ and the details are given below:

- $\mathcal{O}_{\mathsf{KeyGen}}(\cdot,\cdot)$ *is the key generation oracle that allows $\mathcal{A}$ to query key attribute $X \in \mathcal{X}$ and an identifier $id \in \mathcal{I}$, and it runs $\mathcal{RABE}.\mathsf{KeyGen}(msk, st, X, id)$ to return the secret key sk_{id}.*
- $\mathcal{O}_{\mathsf{KeyUpdate}}(\cdot,\cdot)$ *is the key update oracle that allows $\mathcal{A}$ to query key attributes $\bar{X}_t$ associated with the time $t \in \mathcal{T}$, and it runs $\mathcal{RABE}.\mathsf{KeyUpdate}(msk, st, \bar{X}_t, rl)$ to return the key update ku_t.*
- $\mathcal{O}_{\mathsf{Rev}}(\cdot,\cdot)$ *is the revocation oracle that allows $\mathcal{A}$ to query an identifier $id \in \mathcal{I}$ and the time $t \in \mathcal{T}$, and it runs $\mathcal{RABE}.\mathsf{Rev}(rl, id, t)$ to update the revocation list rl.*
- $\mathcal{O}_{\mathsf{DKGen}}(\cdot,\cdot,\cdot)$ *is the decryption key generation oracle that allows $\mathcal{A}$ to query key attributes $(X, \bar{X}) \in \mathcal{X} \times \bar{\mathcal{X}}$, the timestamp $t \in \mathcal{T}$ and an identifier $id \in \mathcal{I}$, and it runs $\mathcal{RABE}.\mathsf{DKGen}(pk, sk_{id}, ku_t)$ to return the decryption key $dk_{id,t}$ if the secret key sk_{id} and the key update ku_t are available. Otherwise, it first runs the key generation oracle and key update oracle to obtain the secret key sk_{id} and the key update ku_t.*

$\mathcal{A}$ is allowed to issue above oracles with the following restrictions:

1. *$\mathcal{O}_{\mathsf{KeyUpdate}}(\cdot,\cdot)$ and $\mathcal{O}_{\mathsf{Rev}}(\cdot,\cdot)$ can be queried at the time t which is greater than or equal to that of all previous queries.*
2. *$\mathcal{O}_{\mathsf{Rev}}(\cdot,\cdot)$ cannot be queried at the time t if $\mathcal{O}_{\mathsf{KeyUpdate}}(\cdot)$ was queried at the time t.*
3. *If $\mathcal{O}_{\mathsf{KeyGen}}(\cdot,\cdot)$ was queried on an identifier $id \in \mathcal{I}$ with key attributes $X \in \mathcal{X}$ s.t. $R(X, Y^*) = 1$, then $\mathcal{O}_{\mathsf{Rev}}(\cdot,\cdot)$ must be queried on this identifier id at the time $t \leq t^*$.*
4. *$\mathcal{O}_{\mathsf{DKGen}}(\cdot,\cdot)$ cannot be queried on any identifier $id \in \mathcal{I}$ with the key attributes $X \in \mathcal{X}$ s.t. $R(X, Y^*) = 1$ at the challenge time t^* or any identifier $id \in \mathcal{I}$ has been revoked.*

An RABE scheme is said to be sIND-CPA *secure if for any probabilistic polynomial time adversary $\mathcal{A}$, the following advantage is negligible:*

$$\mathbf{Adv}^{\mathsf{sIND\text{-}CPA}}_{\mathcal{A},\mathcal{RABE}}(\lambda) = \left| \Pr[\mathbf{Exp}^{\mathsf{sIND\text{-}CPA}}_{\mathcal{A},\mathcal{RABE}}(\lambda) = 1] - 1/2 \right|.$$

3.3 Definition of DP-ABE

Definition 7 (DP-ABE). *Dual-Policy Attribute-Based Encryption $\mathcal{DP}$ with key attributes $(X, \bar{X}) \in \mathcal{X} \times \bar{\mathcal{X}}$ that support ciphertext attributes $(Y, \bar{Y}) \in \mathcal{Y} \times \bar{\mathcal{Y}}$ and the message space $\mathcal{M}$ consists of following four algorithms:*

- $\mathcal{DP}.\mathsf{Setup}(\lambda) \rightarrow (pk, msk)$*: The probabilistic setup algorithm takes as input the security parameter $\lambda \in \mathbb{N}$, and outputs the public key pk and the master secret key msk.*
- $\mathcal{DP}.\mathsf{KeyGen}(msk, X, \bar{X}) \rightarrow sk_{X,\bar{X}}$*: The key generation algorithm takes as input the master secret key msk, the key attributes $(X, \bar{X}) \in \mathcal{X} \times \bar{\mathcal{X}}$, and outputs the secret key $sk_{X,\bar{X}}$.*
- $\mathcal{DP}.\mathsf{Enc}(pk, Y, \bar{Y}, m) \rightarrow c_{Y,\bar{Y}}$*: The encryption algorithm takes as input the public key pk, the ciphertext attributes $(Y, \bar{Y}) \in \mathcal{Y} \times \bar{\mathcal{Y}}$ and the message $m \in \mathcal{M}$, and outputs the ciphertext $c_{Y,\bar{Y}}$.*
- $\mathcal{DP}.\mathsf{Dec}(sk_{X,\bar{X}}, c_{Y,\bar{Y}}) \rightarrow m$*: The decryption algorithm takes as input the secret key $sk_{X,\bar{X}}$ and the ciphertext $c_{Y,\bar{Y}}$, and outputs the message $m \in \mathcal{M}$.*

Definition 8 (Correctness of Parallel DP-ABE). *Let $\mathcal{PDP}$ denote a parallel DP-ABE scheme. The consistency condition requires for all $\lambda \in \mathbb{N}$, the public key pk and the master secret key msk output by setup algorithm, $m \in \mathcal{M}$ and $\big(R(X,Y) \vee \bar{R}(\bar{X},\bar{Y})\big) = 1$, we then have*

$$\mathcal{PDP}.\mathsf{Dec}\left(sk_{X,\bar{X}}, \mathcal{PDP}.\mathsf{Enc}(pk, Y, \bar{Y}, m)\right) = m.$$

Definition 9 (Correctness of Sequential DP-ABE). *Let $\mathcal{SDP}$ denote a sequential DP-ABE scheme. The consistency condition requires for all $\lambda \in \mathbb{N}$, the public key pk and the master secret key msk output by setup algorithm, $m \in \mathcal{M}$, and $R(X,Y) = \bar{R}(\bar{X},\bar{Y}) = 1$, we then have*

$$\mathcal{SDP}.\mathsf{Dec}\left(sk_{X,\bar{X}}, \mathcal{SDP}.\mathsf{Enc}(pk, Y, \bar{Y}, m)\right) = m.$$

3.4 Security Model of DP-ABE

Definition 10 (sIND-CPA in Parallel DP-ABE). *A parallel DP-ABE $\mathcal{PDP}$ consists of four algorithms in above. For an adversary $\mathcal{A}$, we define the following experiment:*

$$
\begin{aligned}
&\mathbf{Exp}^{\mathsf{sIND\text{-}CPA}}_{\mathcal{A},\mathcal{PDP}}(\lambda)\\
&\quad (Y^*, \bar{Y}^*) \leftarrow \mathcal{A}(\lambda);\\
&\quad (pk, msk) \leftarrow \mathcal{PDP}.\mathsf{Setup}(\lambda);\\
&\quad (m_0, m_1) \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{KeyGen}}}(pk);\\
&\quad b \leftarrow \{0,1\};\\
&\quad c^* \leftarrow \mathcal{PDP}.\mathsf{Enc}(pk, Y^*, \bar{Y}^*, m_b);\\
&\quad b' \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{KeyGen}}}(c^*);\\
&\quad \textit{If } b = b' \textit{ return } 1 \textit{ else return } 0.
\end{aligned}
$$

$\mathcal{O}_{\mathsf{KeyGen}}(\cdot,\cdot)$ *is the key generation oracle that allows* $\mathcal{A}$ *to query on any key attributes* $X, \bar{X} \in \mathcal{X} \times \bar{\mathcal{X}}$ *s.t.* $R(X, Y^*) = \bar{R}(\bar{X}, \bar{Y}^*) = 0$*, and returns the secret key* $sk_{X,\bar{X}}$ *by running* $\mathcal{PDP}.\mathsf{KeyGen}(msk, X, \bar{X})$.
A parallel DP-ABE is said to be sIND-CPA *secure if for any probabilistic polynomial time adversary* $\mathcal{A}$*, the following advantage is negligible:*

$$\mathbf{Adv}^{\mathsf{sIND\text{-}CPA}}_{\mathcal{A},\mathcal{PDP}}(\lambda) = \big|\Pr[\mathbf{Exp}^{\mathsf{sIND\text{-}CPA}}_{\mathcal{A},\mathcal{PDP}}(\lambda) = 1] - 1/2\big|.$$

Definition 11 (sIND-CPA in Sequential DP-ABE). *A sequential DP-ABE* $\mathcal{SDP}$ *consists of four algorithms in above. For an adversary* $\mathcal{A}$*, we define the following experiment:*

$$
\begin{aligned}
&\mathbf{Exp}^{\mathsf{sIND\text{-}CPA}}_{\mathcal{A},\mathcal{SDP}}(\lambda)\\
&\quad (Y^*, \bar{Y}^*) \leftarrow \mathcal{A}(\lambda);\\
&\quad (pk, msk) \leftarrow \mathcal{SDP}.\mathsf{Setup}(\lambda);\\
&\quad (m_0, m_1) \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{KeyGen}}}(pk);\\
&\quad b \leftarrow \{0,1\};\\
&\quad c^* \leftarrow \mathcal{SDP}.\mathsf{Enc}(pk, Y^*, \bar{Y}^*, m_b);\\
&\quad b' \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{KeyGen}}}(c^*);\\
&\quad \textit{If } b = b' \textit{ return } 1 \textit{ else return } 0.
\end{aligned}
$$

$\mathcal{O}_{\mathsf{KeyGen}}(\cdot,\cdot)$ *is the key generation oracle that allows* $\mathcal{A}$ *to query on any key attributes* $X, \bar{X} \in \mathcal{X} \times \bar{\mathcal{X}}$ *except* $R(X, Y^*) = \bar{R}(\bar{X}, \bar{Y}^*) = 1$*, and returns the secret key* $sk_{X,\bar{X}}$ *by running* $\mathcal{SDP}.\mathsf{KeyGen}(msk, X, \bar{X})$.
A sequential DP-ABE is said to be sIND-CPA *secure if for any probabilistic polynomial time adversary* $\mathcal{A}$*, the following advantage is negligible:*

$$\mathbf{Adv}^{\mathsf{sIND\text{-}CPA}}_{\mathcal{A},\mathcal{SDP}}(\lambda) = \big|\Pr[\mathbf{Exp}^{\mathsf{sIND\text{-}CPA}}_{\mathcal{A},\mathcal{SDP}}(\lambda) = 1] - 1/2\big|.$$

4 Proposed Schemes

4.1 Generic Construction of Revocable ABE

Let $\mathcal{ETC}$ and $\mathcal{ETC}_t$ are ElGamal type cryptosystems. The generic construction of RABE $\mathcal{RABE}$ are described as follows.

- $\mathcal{RABE}.\mathsf{Setup}(\lambda)$: The setup algorithm initializes an empty revocation list $rl \leftarrow \emptyset$ and a state based on the binary tree BT with $\mathcal{N}$ leaf nodes, where $\mathcal{N}$ is the number of system users. The algorithm follows $\mathcal{SDP}.\mathsf{Setup}(\lambda)$ to generate the public key pk and the master secret key msk.
- $\mathcal{RABE}.\mathsf{KeyGen}(msk, st, X, id)$: The key generation algorithm chooses an unassigned leaf node from the binary tree BT and stores id in this node. For each node $\theta \in \mathsf{Path}(id)$:
  - Fetch α_θ from the node θ. If α_θ is not available, it randomly chooses $\alpha_\theta \in \mathbb{Z}_p$, and updates the state $st \leftarrow st \cup (\theta, \alpha_\theta)$.
  - Run $\mathcal{ETC}.\mathsf{KeyGen}(\alpha_\theta, X) \rightarrow sk_\theta$.

  The key generation algorithm returns the secret key $sk_{id} = \{sk_\theta\}_{\theta \in \mathsf{Path}(id)}$ and the updated state st.
- $\mathcal{RABE}.\mathsf{KeyUpdate}(msk, st, \bar{X}_t, rl) \rightarrow ku_t$: Pases $\bar{X}_t$ is the key attributes based on the timestamp $t \in \mathcal{T}$. For each node $\theta \in \mathsf{KUNodes}(st, rl, t)$:
  - Fetch α_θ (α_θ always predefined in the key generation algorithm).
  - Run $\mathcal{ETC}_t.\mathsf{KeyGen}(\alpha - \alpha_\theta, \bar{X}_t) \rightarrow sk_{t,\theta}$, where α is the master secret key.

  The key update algorithm returns $ku_t = \{sk_{t,\theta}\}_{\theta \in \mathsf{KUNodes}(st,rl,t)}$.
- $\mathcal{RABE}.\mathsf{DKGen}(pk, sk_{id}, ku_t) \rightarrow dk_{id,t}$: Let I and J denote sets $\mathsf{Path}(id)$ and $\mathsf{KUNodes}(st, rl, t)$, respectively. For $\theta \in \mathsf{I} \cap \mathsf{J}$, the algorithm chooses a serial of random values to re-randomize the keys $(sk_\theta, sk_{t,\theta})$ and returns the decryption key $dk_{id} = (sk_\theta, sk_{t,\theta})$.
- $\mathcal{RABE}.\mathsf{Enc}(pk, Y, \bar{Y}_t, m) \rightarrow c_{Y,\bar{Y}_t}$: Same as $\mathcal{SDP}.\mathsf{Enc}(pk, Y, \bar{Y}, m)$.
- $\mathcal{RABE}.\mathsf{Dec}(pk, dk_{id,t}, c_{Y,\bar{Y}_t}) \rightarrow m$: Same as $\mathcal{SDP}.\mathsf{Dec}(dk_{id,t}, c_{Y,\bar{Y}_t})$.
- $\mathcal{RABE}.\mathsf{Rev}(rl, id, t) \rightarrow rl$: The revocation algorithm returns the revocation list rl as $rl \leftarrow rl \cup (id, t)$.

Theorem 1. *If the underlying ElGamal type cryptosystems $\mathcal{ETS}_1$ and $\mathcal{ETS}_2$ are secure, the proposed generic construction is secure*[2].

4.2 Generic Construction of Parallel DP-ABE

Let $\mathcal{ETC}_{\mathsf{kp}}$ and $\mathcal{ETC}_{\mathsf{cp}}$ are ElGamal type cryptosystems based on KP-ABE and CP-ABE, respectively. The generic construction of parallel DP-ABE $\mathcal{PDP}$ are described as follows.

- $\mathcal{PDP}.\mathsf{Setup}(\lambda)$: The setup algorithm runs

$$\begin{cases} \mathcal{ETC}_{\mathsf{kp}}.\mathsf{Init}(\lambda) \rightarrow pp_{\mathsf{kp}}, or \\ \mathcal{ETC}_{\mathsf{cp}}.\mathsf{Init}(\lambda) \rightarrow pp_{\mathsf{cp}}. \end{cases}$$

  to obtain the description of bilinear group as the public parameter pp, where $pp_{\mathsf{kp}} = pp_{\mathsf{cp}} = \mathcal{G}(\lambda)$ by the definition of ElGamal type cryptosystem. The algorithm also runs

$$\begin{cases} \mathcal{ETC}_{\mathsf{kp}}.\mathsf{Setup}(pp) \rightarrow (pk_{\mathsf{kp}}, msk_{\mathsf{kp}}), and \\ \mathcal{ETC}_{\mathsf{cp}}.\mathsf{Setup}(pp) \rightarrow (pk_{\mathsf{cp}}, msk_{\mathsf{cp}}). \end{cases}$$

[2] Please contact the authors for the formal security proofs of Theorem 1 to 3.

to obtain the master secret key α, where $msk_{\mathsf{kp}} = msk_{\mathsf{cp}} = \alpha$ by the definition of ElGamal type cryptosystem. The setup algorithm outputs

$$pk = (pp, pk_{\mathsf{kp}}, pk_{\mathsf{cp}}), \quad msk = (\alpha).$$

– $\mathcal{PDP}.\mathsf{KeyGen}(msk, X, \bar{X})$: Parse X is the access structure in KP-ABE and $\bar{X}$ is attribute set in CP-ABE. The key generation algorithm runs

$$\begin{cases} \mathcal{ETC}_{\mathsf{kp}}.\mathsf{KeyGen}(\alpha, X) \to sk_X, and \\ \mathcal{ETC}_{\mathsf{cp}}.\mathsf{KeyGen}(\alpha, \bar{X}) \to sk_{\bar{X}}. \end{cases}$$

The key generation algorithm outputs the secret key $sk_{X,\bar{X}} = (sk_X, sk_{\bar{X}})$.

– $\mathcal{PDP}.\mathsf{Enc}(pk, Y, \bar{Y}, m)$: Parse Y is the attribute set in KP-ABE and $\bar{Y}$ is the access structure in CP-ABE. The encryption algorithm runs

$$\begin{cases} \mathcal{ETC}_{\mathsf{kp}}.\mathsf{Enc}(pk_{\mathsf{kp}}, Y, m) \to c_Y, and \\ \mathcal{ETC}_{\mathsf{cp}}.\mathsf{Enc}(pk_{\mathsf{cp}}, \bar{Y}, m) \to c_{\bar{Y}}. \end{cases}$$

By the definition of ElGamal type cryptosystem, we have

$$c_Y = (c_Y^{(0)}, c_Y^{(1)}, c_Y^{(2)}) \; and \; c_{\bar{Y}} = (c_{\bar{Y}}^{(0)}, c_{\bar{Y}}^{(1)}, c_{\bar{Y}}^{(2)}),$$

where $c_Y^{(0)} = c_{\bar{Y}}^{(0)} = m \cdot e(g,g)^{\alpha s}$ and $c_Y^{(1)} = c_{\bar{Y}}^{(1)} = g^s$. The encryption algorithm outputs the ciphertext $c_{Y,\bar{Y}} = (m \cdot e(g,g)^{\alpha s}, g^s, c_Y^{(2)}, c_{\bar{Y}}^{(2)})$.

– $\mathcal{PDP}.\mathsf{Dec}(sk_{X,\bar{X}}, c_{Y,\bar{Y}})$: The decryption algorithm runs

$$\begin{cases} \mathcal{ETC}_{\mathsf{kp}}.\mathsf{Dec}(pk_{\mathsf{kp}}, sk_X, c_Y) \to m \text{ if } R(X,Y) = 1, \\ \mathcal{ETC}_{\mathsf{cp}}.\mathsf{Dec}(pk_{\mathsf{cp}}, sk_{\bar{X}}, c_{\bar{Y}}) \to m \text{ if } \bar{R}(\bar{X}, \bar{Y}) = 1. \end{cases}$$

The decryption algorithm returns the message m.

Theorem 2. *If the underlying ElGamal type cryptosystems $\mathcal{ETC}_{\mathsf{kp}}$ and $\mathcal{ETC}_{\mathsf{cp}}$ are secure, the proposed generic construction of parallel DP-ABE is secure.*

4.3 Generic Construction of Sequential DP-ABE

Let $\mathcal{ETC}_{\mathsf{kp}}$ and $\mathcal{ETC}_{\mathsf{cp}}$ are ElGamal type cryptosystems based on KP-ABE and CP-ABE, respectively. The generic construction of parallel DP-ABE $\mathcal{SDP}$ are described as follows.

– $\mathcal{SDP}.\mathsf{Setup}(\lambda)$: Same as $\mathcal{PDP}.\mathsf{Setup}(\lambda)$.

– $\mathcal{SDP}.\mathsf{KeyGen}(msk, X, \bar{X})$: Parse X is the access structure in KP-ABE and $\bar{X}$ is attribute set in CP-ABE. The key generation algorithm randomly picks $\alpha' \in \mathbb{Z}_p$ and runs

$$\begin{cases} \mathcal{ETC}_{\mathsf{kp}}.\mathsf{KeyGen}(\alpha', X) \to sk_X, and \\ \mathcal{ETC}_{\mathsf{cp}}.\mathsf{KeyGen}(\alpha - \alpha', \bar{X}) \to sk_{\bar{X}}. \end{cases}$$

The key generation algorithm outputs the secret key $sk_{X,\bar{X}} = (sk_X, sk_{\bar{X}})$.

– $\mathcal{SDP}.\mathsf{Enc}(pk, Y, \bar{Y}, m)$: Same as $\mathcal{PDP}.\mathsf{Enc}(pk, Y, \bar{Y}, m)$.
– $\mathcal{SDP}.\mathsf{Dec}(sk_{X,\bar{X}}, c_{Y,\bar{Y}})$: The decryption algorithm runs the sub-decryption algorithms

$$\begin{cases} \mathcal{D}_{\mathsf{kp}}(sk_X, g^s, c_Y^{(2)}) \to e(g,g)^{\alpha' s}, \\ \mathcal{D}_{\mathsf{cp}}(sk_{\bar{X}}, g^s, c_{\bar{Y}}^{(2)}) \to e(g,g)^{(\alpha-\alpha')s}. \end{cases}$$

The decryption algorithm outputs the message $m = m \cdot e(g,g)^{\alpha s}/e(g,g)^{\alpha s}$.

Theorem 3. *If the underlying ElGamal type cryptosystems $\mathcal{ETC}_{\mathsf{kp}}$ and $\mathcal{ETC}_{\mathsf{cp}}$ are secure, the proposed generic construction of parallel DP-ABE is secure.*

5 Instantiations Based on ElGamal Type Cryptosystem

5.1 Instantiations of RABE

By applying the generic construction in Sect. 4.1, we can build the concrete instantiation of key-policy RABE and ciphertext RABE, and even revocable DP-ABE by dividing the master secret key into three pieces for $(X, \bar{X}, t)$, where $(X, \bar{X})$ are key attributes in DP-ABE and t is for managing user revocation. There are many concrete RABE schemes based on ElGamal type schemes. For example, the RABE with decryption key exposure resistance [31] are based on [14], and [24], the KP-ABE with efficient revocation mechanism and decryption key exposure resistance [32] are based on [14] and [28], and the CP-ABE with efficient revocation mechanism and decryption key exposure resistance [33] are based on [24] and [28]. We omit the detailed construction here since our paper focus on argue that any ElGamal type scheme as in Definition 3 can be used to build secure RABE schemes and dual-policy ABE, respectively.

5.2 An Instantiation of Parallel DP-ABE

By applying the generic construction in Sect. 4.2, we give an instantiation of parallel DP-ABE based on [14] and [24] as follows.

– $\mathcal{PDP}.\mathsf{Setup}(\lambda)$: Run $\mathcal{G}(\lambda)$ to obtain $(p, g, \mathbb{G}, \mathbb{G}_T)$. Pick $u, h, w, v, \{t_i\}_{i\in[n+1]} \in \mathbb{G}$ and $\alpha \in \mathbb{Z}_p$. Output

$$pk = (p, g, \mathbb{G}, \mathbb{G}_T, e(g,g)^{\alpha}, u, h, w, v, \{t_i\}_{i\in[n+1]}), \quad msk = (\alpha).$$

– $\mathcal{PDP}.\mathsf{KeyGen}(msk, X, \bar{X})$: Parse $X = (\mathbb{M}_{\mathsf{kp}}, \rho_{\mathsf{kp}})$ and $\bar{X} = (A_{\mathsf{cp}}^{(1)}, ..., A_{\mathsf{cp}}^{(k_{\mathsf{cp}})})$. Compute $sk_X = (\{g^{\mathbb{M}_{\mathsf{kp},i}\vec{u}_i} T(i)^{r_i}, g^{r_i}\}_{i\in[d_{\mathsf{kp}}]})$, where $\mathbb{M}_{\mathsf{kp}}$ has d_{kp} rows and l_{kp} columns, and $\vec{1}\cdot\vec{u} = \alpha$. Compute $sk_{\bar{X}} = (g^{\alpha} w^r, g^r, \{g^{r_j}, (u^{A_{\mathsf{cp}}^{(j)}} h)^{r_j} v^{-r}\}_{j\in[k_{\mathsf{cp}}]})$, where $r, \{r_j\}_{j\in[k_{\mathsf{cp}}]} \in \mathbb{Z}_p$. Output $sk_{X,\bar{X}} = (sk_X, sk_{\bar{X}})$.
– $\mathcal{PDP}.\mathsf{Enc}(pk, Y, \bar{Y}, m)$: Parse $Y = (A_{\mathsf{kp}}^{(1)}, A_{\mathsf{kp}}^{(2)}, ..., A_{\mathsf{kp}}^{(k_{\mathsf{kp}})})$ and $\bar{Y} = (\mathbb{M}_{\mathsf{cp}}, \rho_{\mathsf{cp}})$. Pick $\vec{u} = (s, \vec{u}_2, ..., \vec{u}_l) \in \mathbb{Z}_p^l$ and compute

$$c_Y^{(3)} = (\{T(i)^s\}_{i\in[k_{\mathsf{kp}}]}), \quad c_{\bar{Y}}^{(3)} = (\{w^{\mathbb{M}_{\mathsf{cp},j}\vec{u}_j} v^{\phi_j}, (u^{\rho(i)} h)^{\phi_i}, g^{\phi_j}\}_{j\in[d_{\mathsf{cp}}]}),$$

where $\mathbb{M}_{\mathsf{cp}}$ has d_{cp} rows and l_{cp} columns, $\{\phi_j\}_{j\in[d_{\mathsf{cp}}]} \in \mathbb{Z}_p$. Output

$$c_{Y,\bar{Y}} = (m \cdot e(g,g)^{\alpha s}, g^s, c_Y^{(3)}, c_{\bar{Y}}^{(3)}).$$

– $\mathcal{PDP}.\mathsf{Dec}(sk_{X,\bar{X}}, c_{Y,\bar{Y}})$: If $R(X,Y)=1$, there exist $I:\{i:\rho_{\mathsf{kp}}(i)\in\mathcal{S}_{\mathsf{kp}}\}$ and take $\vec{u}$ s.t. $\sum_{i\in I}\mathbb{M}_{\mathsf{kp},i}\vec{u}_i = \vec{1}$. Compute

$$\prod_{i\in I}\left(\frac{e(g^{\mathbb{M}_{\mathsf{kp},i}\vec{u}_i}T(i)^{r_i}, g^s)}{e(T(i)^s, g^{r_i})}\right)^{\vec{u}_i} = e(g,g)^{\alpha s}.$$

If $\bar{R}(\bar{X},\bar{Y})=1$, there exist $J=\{j:\rho_{\mathsf{cp}}(j)\in\mathcal{S}_{\mathsf{cp}}\}$ and take $\vec{u}$ s.t. $\sum_{j\in J}\mathbb{M}_{\mathsf{cp},j}\vec{u}_j = \vec{1}$. Also, compute

$$\prod_{j\in J}\frac{e(g^s, g^\alpha w^r)}{\left(e(w^{\mathbb{M}_{\mathsf{cp},j}\vec{u}_j}v^{\phi_j}, g^r)e((u^{\rho(i)}h)^{\phi_i}, g^{r_j})e(g^{\phi_j}, (u^{A_{\mathsf{cp}}^{(j)}}h)^{r_j}v^{-r})\right)^{\vec{u}_j}} = e(g,g)^{\alpha s}.$$

Output $m = m\cdot e(g,g)^{\alpha s}/e(g,g)^{\alpha s}$.

5.3 An Instantiation of Sequential DP-ABE

By applying the generic construction in Sect. 4.3, we give an instantiation of sequential DP-ABE based on [14] and [24] as follows.

– $\mathcal{SDP}.\mathsf{Setup}(\lambda)$: Same as $\mathcal{PDP}.\mathsf{Setup}(\lambda)$.
– $\mathcal{SDP}.\mathsf{KeyGen}(msk, X, \bar{X})$: Parse $X=(\mathbb{M}_{\mathsf{kp}},\rho_{\mathsf{kp}})$ and $\bar{X}=(A_{\mathsf{cp}}^{(1)},...,A_{\mathsf{cp}}^{(k_{\mathsf{cp}})})$. Compute $sk_X = (\{g^{\mathbb{M}_{\mathsf{kp},i}\vec{u}_i}T(i)^{r_i}, g^{r_i}\}_{i\in[d_{\mathsf{kp}}]})$, where $\mathbb{M}_{\mathsf{kp}}$ has d_{kp} rows and l_{kp} columns, $\alpha', \{r_i\}_{i\in[d_{\mathsf{kp}}]} \in \mathbb{Z}_p$ and $\vec{1}\cdot\vec{u}=\alpha'$. Also, compute $sk_{\bar{X}} = (g^{\alpha-\alpha'}w^r, g^r, \{g^{r_j}, (u^{A_{\mathsf{cp}}^{(j)}}h)^{r_j}v^{-r}\}_{j\in[k_{\mathsf{cp}}]})$, where $r, \{r_j\}_{j\in[k_{\mathsf{cp}}]} \in \mathbb{Z}_p$. Output $sk_{X,\bar{X}} = (sk_X, sk_{\bar{X}})$.
– $\mathcal{SDP}.\mathsf{Enc}(pk, Y, \bar{Y}, m)$: Same as $\mathcal{PDP}.\mathsf{Enc}(pk, Y, \bar{Y}, m)$.
– $\mathcal{SDP}.\mathsf{Dec}(sk_{X,\bar{X}}, c_{Y,\bar{Y}})$: If $R(X,Y)=1$, there exist $I:\{i:\rho_{\mathsf{kp}}(i)\in\mathcal{S}_{\mathsf{kp}}\}$ and take $\vec{u}$ s.t. $\sum_{i\in I}\mathbb{M}_{\mathsf{kp},i}\vec{u}_i = \vec{1}$. Compute

$$\prod_{i\in I}\left(\frac{e(g^{\mathbb{M}_{\mathsf{kp},i}\vec{u}_i}T(i)^{r_i}, g^s)}{e(T(i)^s, g^{r_i})}\right)^{\vec{u}_i} = e(g,g)^{\alpha' s}.$$

If $R(\bar{X},\bar{Y})=1$, there exist $J=\{j:\rho_{\mathsf{cp}}(j)\in\mathcal{S}_{\mathsf{cp}}\}$ and take $\vec{u}$ s.t. $\sum_{j\in J}\mathbb{M}_{\mathsf{cp},j}\vec{u}_j = \vec{1}$. Also, compute

$$\prod_{j\in J}\frac{e(g^s, g^{\alpha-\alpha'}w^r)}{\left(e(w^{\mathbb{M}_{\mathsf{cp},j}\vec{u}_j}v^{\phi_j}, g^r)e((u^{\rho(i)}h)^{\phi_i}, g^{r_j})e(g^{\phi_j}, (u^{A_{\mathsf{cp}}^{(j)}}h)^{r_j}v^{-r})\right)^{\vec{u}_j}} = e(g,g)^{(\alpha-\alpha')s}.$$

Output $m = m\cdot e(g,g)^{\alpha s}/(e(g,g)^{\alpha' s}\cdot e(g,g)^{(\alpha-\alpha')s})$.

6 Efficiency Analysis

To our knowledge, only few literature investigate DP-ABE [2,4]. Compared with our proposed scheme, as shown in Table 1, our schemes have better performances than AI09 [2] and less efficient then AY15 [4]. However, to achieve adaptive security, AY15 is in the composite-order group which is less efficient[3] since it will incur heavy workload to process data, even transmission bandwidth. Our proposed scheme applies prime-order group and has the same complexity to the existing DP-ABE schemes except for the space complexity of system parameter. In our scheme, the component of KP-ABE is based on [14] and the part of CP-ABE is based [24], where [14] has the linear space complexity on the public parameter and [24] has the constant-size public parameter. Hence, our scheme only has better space complexity on the system parameter than AI09. Although AY15 has the constant-size public parameter, the composite-order group will lead to a heavy workload.

Table 1. Theoretical analysis of DP-ABE scheme

	Space complexity			Computational complexity	
	Parameter	Secret key	Ciphertext	Encryption	Decryption
AI09 [2]	$\mathcal{O}(m+n)$	$\mathcal{O}(\mathcal{X}+\bar{\mathcal{X}})$	$\mathcal{O}(\mathcal{Y}+\bar{\mathcal{Y}})$	$\mathcal{O}(\mathcal{Y}+\bar{\mathcal{Y}})$	$\mathcal{O}(\mathcal{X}+\bar{\mathcal{X}})$
AY15 [4]	$\mathcal{O}(1)$	$\mathcal{O}(\mathcal{X}+\bar{\mathcal{X}})$	$\mathcal{O}(\mathcal{Y}+\bar{\mathcal{Y}})$	$\mathcal{O}(\mathcal{Y}+\bar{\mathcal{Y}})$	$\mathcal{O}(\mathcal{X}+\bar{\mathcal{X}})$
SDP-ABE	$\mathcal{O}(m)$	$\mathcal{O}(\mathcal{X}+\bar{\mathcal{X}})$	$\mathcal{O}(\mathcal{Y}+\bar{\mathcal{Y}})$	$\mathcal{O}(\mathcal{Y}+\bar{\mathcal{Y}})$	$\mathcal{O}(\mathcal{X}+\bar{\mathcal{X}})$
PDP-ABE	$\mathcal{O}(m)$	$\mathcal{O}(\mathcal{X}+\bar{\mathcal{X}})$	$\mathcal{O}(\mathcal{Y}+\bar{\mathcal{Y}})$	$\mathcal{O}(\mathcal{Y}+\bar{\mathcal{Y}})$	$\mathcal{O}(\mathcal{X}+\bar{\mathcal{X}})$

m denotes the maximum size of attribute set allowed to be assigned to a key;
n is the maximum size of attribute set to be associated with a ciphertext;
$\mathcal{X}$ and $\bar{\mathcal{X}}$ represent the size of attributes and policies assigned to a key;
$\mathcal{Y}$ and $\bar{\mathcal{Y}}$ represent the size of policies and attributes assigned to a ciphertext.

For experimental analysis, we focus on evaluating AI09 and our schemes since the AY15 based on the inefficient composite-order group. Our experimental simulation was performed on a PC running 64-bit Windows 10 with 3.60 GHz Intel(R) Core(TM) i7-4790 CPU and 24 GB memory. We have implemented AI09 and our schemes in Java using JPBE library [19] with Type A elliptic curve and symmetric pairing setting from "a properties" provided by JPBE library. Hence, our scheme, p is a 160-bit prime number, and elements in $\mathbb{G}$ and $\mathbb{G}_T$ have 512-bit and 1024-bit, respectively. The experimental results are presented in Fig. 3.

Figure 3a presents the experimental performances of the system initialization by increasing the maximum number of attribute set allowed to be assigned to a key and a ciphertext. Our proposed schemes are much more efficient than AI09, which only take half of the computational time in AI09. Figure 3b performs the

[3] Composite-order group has a much bigger size than the prime-order group. Specifically, the composite-order group needs 1024 bits if the prime-order group requires 160 bits (discrete log vs. factoring).

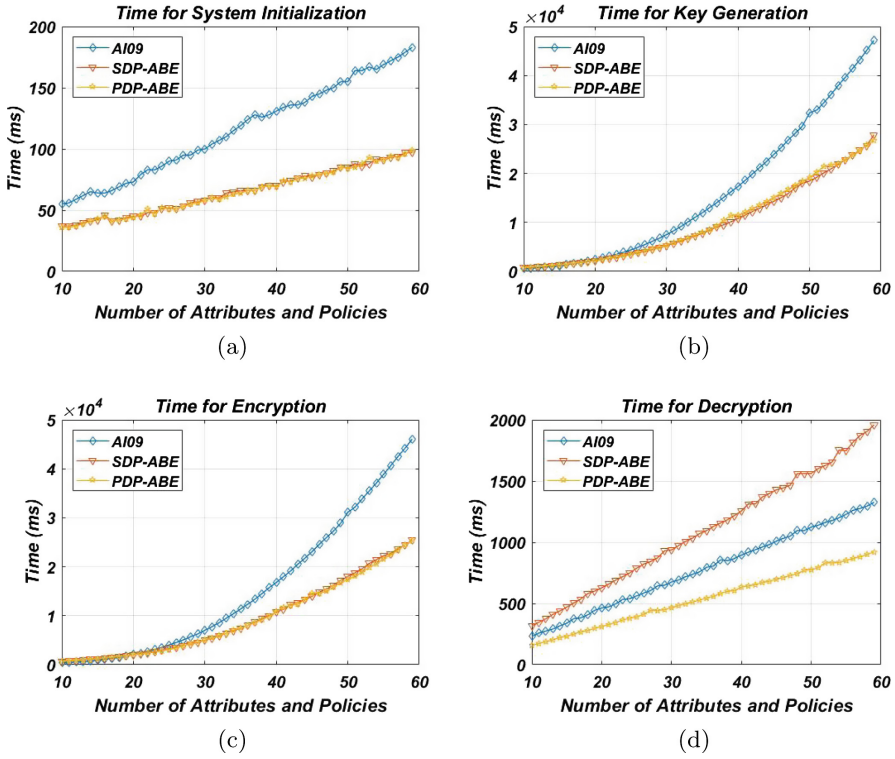

Fig. 3. Experimental performance

ms key generation, the tendency is continually increasing based on the improvement of the maximum number of attribute set and policies allowed to be assigned to keys, and our proposed schemes have the lower growth rate. Figure 3c demonstrates the performances of encryption, the tendency of encryption is similar to the experimental result in the key generation. Figure 3d presents the results of decryption. Our PDP-ABE has a better performance than others since it only requires one of key-policy and ciphertext-policy to process the decryption algorithm, which takes half of the computational cost in SDP-ABE.

Overall, the results are similar to what we expected performances in Table 1. Therefore, our scheme has better performance than the existing DP-ABE based on the prime-order group.

7 Conclusion

We resisted IBE and ABE schemes and presented a new cryptographic primitive called ElGamal type cryptosystem. ElGamal type cryptosystem is a useful primitive for designing a variety of ABE schemes. In this paper, we present generic constructions of RABE with decryption key exposure resistance and DP-ABE

with parallel and sequential settings and the corresponding security proofs. We also provide instantiations of these schemes and the experimental data of DP-ABE to demonstrate high performances of our proposed schemes.

Acknowledgment. This research is supported by the National Natural Science Foundation of China under Grant Nos. U1804263 and 61702105, the Key Research and Development Program of Shaanxi under Grant 2019KW-053, and the New Star Team of Xi'an University of Posts and Telecommunications under Grant 2016-02.

References

1. Abdalla, M., Catalano, D., Dent, A.W., Malone-Lee, J., Neven, G., Smart, N.P.: Identity-based encryption gone wild. In: Bugliesi, M., Preneel, B., Sassone, V., Wegener, I. (eds.) ICALP 2006. LNCS, vol. 4052, pp. 300–311. Springer, Heidelberg (2006). https://doi.org/10.1007/11787006_26
2. Attrapadung, N., Imai, H.: Dual-policy attribute based encryption. In: Abdalla, M., Pointcheval, D., Fouque, P.-A., Vergnaud, D. (eds.) ACNS 2009. LNCS, vol. 5536, pp. 168–185. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-01957-9_11
3. Attrapadung, N., Libert, B., de Panafieu, E.: Expressive key-policy attribute-based encryption with constant-size ciphertexts. In: Catalano, D., Fazio, N., Gennaro, R., Nicolosi, A. (eds.) PKC 2011. LNCS, vol. 6571, pp. 90–108. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-19379-8_6
4. Attrapadung, N., Yamada, S.: Duality in ABE: converting attribute based encryption for dual predicate and dual policy via computational encodings. In: Nyberg, K. (ed.) CT-RSA 2015. LNCS, vol. 9048, pp. 87–105. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-16715-2_5
5. 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). https://doi.org/10.1007/978-3-540-24632-9_19
6. Bethencourt, J., Sahai, A., Waters, B.: Ciphertext-policy attribute-based encryption. In: IEEE S&P, pp. 321–334 (2007)
7. Boldyreva, A., Goyal, V., Kumar, V.: Identity-based encryption with efficient revocation. In: CCS, pp. 417–426 (2008)
8. Boneh, D., Boyen, X.: Efficient selective-ID secure identity-based encryption without random oracles. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 223–238. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-24676-3_14
9. Boneh, D., Boyen, X.: Secure identity based encryption without random oracles. In: Franklin, M. (ed.) CRYPTO 2004. LNCS, vol. 3152, pp. 443–459. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-28628-8_27
10. Boneh, D., Franklin, M.: Identity-based encryption from the weil pairing. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 213–229. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-44647-8_13
11. Chow, S.S.M.: A framework of multi-authority attribute-based encryption with outsourcing and revocation. In: SACMAT, pp. 215–226 (2016)
12. ElGamal, T.: A public key cryptosystem and a signature scheme based on discrete logarithms. In: Blakley, G.R., Chaum, D. (eds.) CRYPTO 1984. LNCS, vol. 196, pp. 10–18. Springer, Heidelberg (1985). https://doi.org/10.1007/3-540-39568-7_2

13. Gentry, C.: Practical identity-based encryption without random oracles. In: Vaudenay, S. (ed.) EUROCRYPT 2006. LNCS, vol. 4004, pp. 445–464. Springer, Heidelberg (2006). https://doi.org/10.1007/11761679_27
14. Goyal, V., Pandey, O., Sahai, A., Waters, B.: Attribute-based encryption for fine-grained access control of encrypted data. In: CCS, pp. 89–98 (2006)
15. Green, M., Hohenberger, S., Waters, B.: Outsourcing the decryption of ABE ciphertexts. In: USENIX (2011)
16. Lee, K., Choi, S.G., Lee, D.H., Park, J.H., Yung, M.: Self-updatable encryption: time constrained access control with hidden attributes and better efficiency. In: Sako, K., Sarkar, P. (eds.) ASIACRYPT 2013. LNCS, vol. 8269, pp. 235–254. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-42033-7_13
17. Lewko, A., Okamoto, T., Sahai, A., Takashima, K., Waters, B.: Fully secure functional encryption: attribute-based encryption and (hierarchical) inner product encryption. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 62–91. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-13190-5_4
18. Lewko, A., Waters, B.: New techniques for dual system encryption and fully secure HIBE with short ciphertexts. In: Micciancio, D. (ed.) TCC 2010. LNCS, vol. 5978, pp. 455–479. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-11799-2_27
19. Lynn, B.: PBC library manual (2006)
20. Malluhi, Q.M., Shikfa, A., Trinh, V.C.: A ciphertext-policy attribute-based encryption scheme with optimized ciphertext size and fast decryption. In: AsiaCCS, pp. 230–240 (2017)
21. Naor, D., Naor, M., Lotspiech, J.: Revocation and tracing schemes for stateless receivers. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 41–62. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-44647-8_3
22. Ostrovsky, R., Sahai, A., Waters, B.: Attribute-based encryption with non-monotonic access structures. In: CCS pp. 195–203 (2007)
23. Pirretti, M., Traynor, P., McDaniel, P., Waters, B.: Secure attribute-based systems. In: CCS, pp. 99–112 (2006)
24. Rouselakis, Y., Waters, B.: Practical constructions and new proof methods for large universe attribute-based encryption. In: CCS, pp. 463–474 (2013)
25. 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). https://doi.org/10.1007/978-3-642-32009-5_13
26. Sahai, A., Waters, B.: Fuzzy identity-based encryption. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 457–473. Springer, Heidelberg (2005). https://doi.org/10.1007/11426639_27
27. 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). https://doi.org/10.1007/978-3-642-36362-7_14
28. Waters, B.: Efficient identity-based encryption without random oracles. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 114–127. Springer, Heidelberg (2005). https://doi.org/10.1007/11426639_7
29. Waters, B.: Dual system encryption: realizing fully secure IBE and HIBE under simple assumptions. In: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 619–636. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-03356-8_36

30. 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). https://doi.org/10.1007/978-3-642-19379-8_4
31. Shengmin, X., Yang, G., Yi, M.: Revocable attribute-based encryption with decryption key exposure resistance and ciphertext delegation. Inf. Sci. **479**, 116–134 (2019)
32. Shengmin, X., Yang, G., Yi, M., Deng, R.H.: Secure fine-grained access control and data sharing for dynamic groups in the cloud. IEEE Trans. Inf. Forensics Secur. **13**(8), 2101–2113 (2018)
33. Shengmin, X., Yang, G., Yi, M., Liu, X.: A secure IoT cloud storage system with fine-grained access control and decryption key exposure resistance. Futur. Gener. Comput. Syst. **97**, 284–294 (2019)