Fully Secure Ciphertext-Policy Attribute Based Encryption with Security Mediator

Yuechen Chen[1,5], Zoe L. Jiang[1], S.M. Yiu[2], Joseph K. Liu[3], Man Ho Au[4], and Xuan Wang[1,6(✉)]

[1] Harbin Institute of Technology Shenzhen Graduate School, Shenzhen 518055, China
[2] HKSAR, The University of Hong Kong, Hong Kong, China
[3] Institute for Infocomm Research, Singapore, Singapore
[4] University of Wollongong, Wollongong, Australia
[5] Shenzhen Applied Technology Engineering Laboratory for Internet Multimedia Application, Shenzhen, China
[6] Public Service Platform of Mobile Internet Application Security Industry, Shenzhen, China
wangxuan@cs.hitsz.edu.cn

Abstract. Attribute-Based Encryption (ABE) offers fine-grained decryption policy such that users can do decryption if their attributes satisfy the policy. Such flexibility enables it applicable in various applications in government and business. However, there are two issues that should be solved first before it is deployed in practice, namely user revocation and decryption outsourcing. In this paper, we adopt the slightly modified Lewko et al.'s fully-CCA-secure Ciphertext-Policy-ABE (CP-ABE) combining with Boneh et al.'s idea of mediated cryptography to propose a CP-ABE with SEcurity Mediator (SEM) supporting immediate user revocation. At the same time, by the introduce of SEM, we intendedly outsource most of the computation workload in decryption to SEM side and leave only one exponentiation and one division at user side for decryption. It is proved fully-RCCA-CCA-secure in random oracle model.

Keywords: CP-ABE · Decryption outsourcing · Dual encryption system · Security mediator · User revocation

1 Introduction

In a traditional Identity-Based Encryption (IBE) system, data is encrypted by a certain identity which can be decrypted by the corresponding secret key. Ciphertext encrypted by the identity can only be decrypted by the secret key. However, in many cases, it is required for any user (with a certain set of attributes) who satisfies a policy can decrypt the corresponding data. For example, the head agent may specify that people satisfying ((Public Corruption Office

Zoe L. Jiang—Co-corresponding author.

© Springer International Publishing Switzerland 2015
L.C.K. Hui et al. (Eds.): ICICS 2014, LNCS 8958, pp. 274–289, 2015.
DOI: 10.1007/978-3-319-21966-0_20

AND (KNOXVILLE OR SAN FRANCISCO)) OR (MANAGEMENT-LEVEL > 5) OR NAME=CHARLIE) to decrypt documents [4]. We call the encryption scheme achieving the above requirement Attribute-Based Encryption (ABE).

ABE can be divided into Key-Policy ABE (KP-ABE) and Ciphertext-Policy ABE (CP-ABE). In KP-ABE [1,2,6,15], secret key is associated with access policy P. And ciphertext is associated with user's attribute set S. The secret key can decrypt the ciphertext if the attribute set S satisfies the policy P. In CP-ABE, ciphertext is associated with policy P, while secret key is associated with user's attribute set S [4,5,11,13].

ACCESS STRUCTURE. ABE is proposed for fine-grained access control on encrypted data. The simplest access structure is Threshold. The user will wish to encrypt a document to all users that have a certain set of attributes. For example, in a department of computer science, the chairperson might want to encrypt a document to all of its systems faculty on a hiring committee. In this case it would encrypt to the attribute set hiring-committee, faculty, systems. Any user who has these attributes could decrypt the document [1]. Access structure with AND and OR gates can be represented by an access tree using 2 of 2 and 1 of 2 Threshold gates as root nodes, respectively. Recall the head agent case in [4] as an example. Such monotone access structure can be further extended to the non-monotonic structure supporting NOT gate [5,6].

SELECTIVE AND FULLY SECURITY. A security model is called *selective* if the adversary is required to announce the intended target before Setup is executed in the game. This is a limited model as the adversary is not necessary to decide the target at the very beginning of the game. There are a list of ABE schemes achieving selective-attribute security [1,2,4–6,11]. The model can be improved to *fully* secure if such limitation is removed such that the adversary will decide the intended target at any time during the game [13–15].

CPA, CCA AND RCCA. To prove an ABE is Chosen-Plaintext-Attack (CPA) secure, we first construct the security game describing the interaction of an adversary $\mathcal{A}$ and challenger $\mathcal{C}$ (i.e., the attacks $\mathcal{A}$ launches), as well as the goal of $\mathcal{A}$ by successfully guessing b from the challenge ciphertext CT_b. Achieving the goal means $\mathcal{A}$ wins the game. We say an ABE is secure if $\mathcal{A}$ wins the game with negligible probability ϵ. Furthermore, we say an ABE is CPA-secure if $\mathcal{A}$ is not allowed to ask for plaintext of his chosen ciphertext. On the contrary, it is Chosen-Ciphertext-Attack secure(CCA-secure). If $\mathcal{A}$ can launch decryption queries on any chosen ciphertext C except the challenge ciphertext C_b (i.e., $C \neq C_b$).The most recent research on ABE is a fully-CCA-secure CP/KP-ABE with monotonic access structure [13]. Replayable CCA (RCCA) was proposed [24] that allows modification to the ciphertext provided they cannot change the underlying plaintext in a meaningful way. Green et al. [16] proposed to outsource the decryption of ABE which is selective-RCCA secure.

USER REVOCATION. Boldyreva, Goyal and Kumar proposed Identity Based Encryption with Efficient Revocation which supports user revocation in IBE [7] in 2008. Then user revocation has been taken notice in many practical ABE systems. User revocation is an essential mechanism in many group-based appli-

cations, including ABE systems, because users may leave their group. Revocation of any single user would affect others who have the same attribute with him. An intuitive way is to append to each of the attributes a date for when the attribute expires [4]. It not only degrades the security in terms of Backward and Forward Secrecy, but also has the scalability problem [18]. Another way is to use ABE that supports NOT gate [6]. Particularly, one can append a NOT gate of the revoked users identity to the previous formula with an AND gate. However, it is very inefficient and does not support immediate revocation. Yu et al. proposed a CP-ABE scheme with attribute immediate revocation [19]. However, it depends on a curious-but-honest proxy server to update secret keys for all unrevoked users. The third method is to re-encrypt the affected ciphertexts, and then updates the decryption keys for unrevoked users. The last method is to introduce a mediator who maintains a real-time RL, which is used in our construction.

As discussed above, user revocation is one of the obstacles for ABE into practice. The other obstacle is the efficiency of ABE. As most of the ABE decryption algorithms involve several paring operations, it is a significant challenge for users using light-weight devices for decryption. Therefore, an ABE scheme supporting decryption outsourcing would be a better solution.

Mediated cryptography was designed by Boneh et al. [20] as a method to allow immediate revocation of public keys. The basic idea when deployed in ABE is to introduce an on-line SEcurity Mediator, SEM, for the check of user validity. Once SEM is notified that a user is to be revoked, it can stop the decryption by the user immediately. To this end, SEM is given a partial decryption key with a real-time revocation list (RL). All encrypted data will be sent to SEM first for checking the validity of the corresponding user in RL. SEM executes SEM-Decrypt if and only if he passes the check, and sends the partially decrypted data to the user for User-Decrypt [21]. Therefore, deploying SEM between the data owner and the data user is a promising solution for ABE. Ibraimi et al. combined this idea with Bethencourt et al.'s selective-CPA-secure CP-ABE [4] to propose a selective-CPA-secure mediated CP-ABE to support immediate user revocation with application in Personal Health Records (PHR) management [25]. In addition, SEM can act as an outsourcing server to do partial decryption [16]. In other words, if we consciously leverage most of the decryption computation to SEM, we can achieve a practical ABE scheme in User-Decrypt.Therefore, the introduction of SEM can solve both the user revocation and decryption outsourcing problems, which directly makes an ABE with SEM to practical applications. However, with the introduction of SEM (a new role in the security game), it changes the original security model. We also need to prove its security although the underlying ABE has been proved fully-CCA secure.

1.1 Our Approach

We are able to obtain a fully-RCCA-CCA-secure CP-ABE with mediator.

Firstly, to achieve immediate user revocation, we adopt the slightly modified Lewko et al.'s fully-CCA-secure CP-ABE [13] as the ABE building block combining with Boneh et al.'s idea of security mediator [20] to propose a CP-ABE with

mediator (SEM). In this scheme, SEM is given a partial decryption key (SK_M) and a real-time Revocation List (RL). Any ciphertext for user decryption should be sent to SEM for user revocation check first. Only passed ciphertext will be partially decrypted using SEM's SK_M. The output will be further sent to user for final decryption. If some users are revoked, SEM will not send the partial decryption result, named CT_M, to the revoked users.

Secondly, to achieve decryption outsourcing, we employ Green et al.'s idea by moving most of the decryption from user side to SEM side (Suppose SEM has strong computational power in cloud). Specifically, raise user's each part of secret key by $1/z (z \in_R Z_p^*)$ to get the partial decryption key for SEM (SK_M). Redefine user's secret key as $SK = (SK_M, z)$. SEM-Decrypt executed by SEM using SK_M will include all pairing operations. But SEM cannot finally decrypt it due to the absence of z. At last, user is only required to execute 1 exponentiation and 1 division to get the plaintext.

1.2 Related Work

Attribute Based Encryption. Sahai and Waters [1] proposed fuzzy identity-based encryption in 2005, which was also called attribute-based encryption when it is applied in the case that an encrypted document can only be decrypted by the user who have a certain set of attributes. In other words, a message encrypted by a set of attribute S can only be decrypted by a private key for another set of attributes S', if and only if $|S \cap S'| \geq d$, a Threshold.

Goyal et al. [2] classified it into key-policy attribute-based encryption (KP-ABE) and cipher-policy attribute-based encryption (CP-ABE), and further proposed a small universe KP-ABE supporting a more general access structure, *monotonic access tree*. Any policy/formula with AND and OR gates can be transformed to such access structure. It is selective-attribute CPA secure without random oracle. They also proposed a large universe KP-ABE which is selective-attribute CPA secure in random oracle. They for the first time proved that it is CCA secure by leveraging the delegation property of their large universe KP-ABE and applying the method in [3]. In 2007, Bethencourt et al. [4] gave the first construction of CP-ABE with monotonic access tree, which is selective-attribute CPA secure with random oracle. They also argued that with delegation property, their scheme is CCA secure. They implemented the ABE scheme using the Pairing Based Cryptography library [17]. In the same year, Cheung and Newport [5] proposed a CP-ABE scheme with access structures of AND gates on both positive and negative attributes. Ostrovsky et al. [6] extended Goyal et al.'s scheme [2] to support non-monotonic access structure, i.e., the Boolean formula involving AND, OR, NOT, and Threshold operations.

In 2011, Waters [11] proposed a selective-attribute CPA secure CP-ABE under DPBDHE assumption with ciphertext size, encryption, and decryption time scales linearly with the complexity of the access formula. Lewko et al. [14] proposed the first fully secure CP-ABE under 3 assumption by adapting Waters' dual system encryption technique in proof model [12]. Okamoto and Takashima [15] proposed a fully secure KP-ABE under DLIN assumption in the same year.

Yamada et al. [13] clearly defined the two properties, delegatability and verifiability in ABE, and argued that any CPA-secure ABE can be transformed to a CCA-secure ABE as long as it has either property above. They also instantiated the variation of Lewko et al.'s CP-ABE, which is fully CCA-secure.

ABE WITH REVOCATION. Bethencourt et al. [4] and Pirretti et al. [26,27] respectively realized *coarse-grained* attribute revocation by revoking attribute itself using timed rekeying mechanism, which was implemented by setting expiration time on each attribute. Attrapadung and Imai proposed ABE with user revocation [8] within similar method as described in [7]. Liang etc. proposed CP-ABE with revocation and it is proved secure under standard model [9]. Qian and Dong proposed Fully Secure Revocable ABE, combined dual encryption and user revocable ABE together to get fully secure ABE with user revocation [10]. However their schemes don't allow immediate revocation. For *fine-grained* user revocation, Ostrovsky et al. [6] proposed to add conjunctively the AND of negation of revoked user attributes. Yu et al. [19] achieved a CP-ABE scheme with immediate attribute revocation which is selective-CCA-secure. Hur and Noh [18] proposed attribute-based access control with efficient revocation without formal security proof. Sahai et al. [28] proposed a fully secure revocable key-policy ABE scheme without concerning Forward Secrecy. Ibraimi et al. combined Bethencourt et al.'s selective-CPA-secure CP-ABE [4] with Boneh et al.'s mediated cryptography [20] to propose a selective-CPA-secure mediated CP-ABE to support immediate user revocation with application [25].

ABE WITH DECRYPTION OUTSOURCING. Green et al. [16] proposed a new paradigm by outsourcing the main decryption computation workload of Waters' CP-ABE [11], and proved it secure in Replayable CCA model, followed by implementing it using PBC library [17].

2 Preliminaries

We review Bilinear maps, decisional $q-$*parallel* Bilinear Diffie-Hellman Exponent problem and linear secret sharing scheme.

2.1 Bilinear Maps

We review some facts related to groups with efficiently computable bilinear maps in [11] and then give our number theoretic assumptions. Let $\mathbb{G}$ and $\mathbb{G}_T$ be two multiplicative cyclic groups of prime order p. Let g be a generator of $\mathbb{G}$ and e be a bilinear map, $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$. The bilinear map e has the following properties:

1. Bilinearity: for all $u, v \in \mathbb{G}$ and $a, b \in \mathbb{Z}_p$, we have $e(u^a, v^b) = e(u, v)^{ab}$.
2. Non-degeneracy: $e(g, g) \neq 1$.

We say that $\mathbb{G}$ is a bilinear group if the group operation in $\mathbb{G}$ and the bilinear map $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ are both efficiently computable. Notice that the map e is symmetric since $e(g^a, g^b) = e(g, g)^{ab} = e(g^b, g^a)$.

2.2 Decisional Parallel BDHE Assumption

We review the definition of decisional $q-parallel$ Bilinear Diffie-Hellman Exponent problem in [11] as follows. Choose a group $\mathbb{G}$ of prime order p according to the security parameter λ. Let $a, s, b_1, \ldots, b_q \in \mathbb{Z}_p$ be chosen at random and g be a generator of $\mathbb{G}$. If an adversary is given $\boldsymbol{y} =$

$$g, g^s, g^a, \ldots, g^{a^q}, , g^{a^{q+2}}, \ldots, g^{a^{2q}}$$
$$\forall_{1 \le j \le q} g^{s \cdot b_j}, g^{a/b_j}, \ldots, g^{a^q/b_j}, , g^{a^{q+2}/b_j}, \ldots, g^{a^{2q}/b_j}$$
$$\forall_{1 \le j \le q, k \ne j} g^{a \cdot s \cdot b_k/b_j}, \ldots, g^{a^q \cdot s \cdot b_k/b_j},$$

it is hard to distinguish $e(g,g)^{a^{q+1}s} \in \mathbb{G}_T$ from a random element R in $\mathbb{G}_T$.

An algorithm $\mathcal{B}$ that outputs $z \in \{0,1\}$ has advantage ϵ in solving decisional *q-parallel* BDHE in $\mathbb{G}$ if

$$|Pr[\mathcal{B}(\boldsymbol{y}, T = e(g,g)^{a^{q+1}s}) = 0] - Pr[\mathcal{B}(\boldsymbol{y}, T = R) = 0]| \ge \epsilon$$

Definition 1. *We say that the (decision) q-parallel-BDHE assumption holds if no polynomial time algorithm $\mathcal{B}$ has a non-negligible advantage in solving the decisional q-parallel BDHE problem.*

2.3 Linear Secret Sharing Schemes

We review the definition of linear secret sharing scheme (LSSS) in [11] as follows.

Definition 2. (Linear Secret-Sharing Schemes (LSSS)) *A secret-sharing scheme over a set of parties $\mathcal{P}$ is called linear (over $\mathbb{Z}_p$) if*

1. *The shares for each party form a vector over $\mathbb{Z}_p$.*
2. *There exists a matrix an M with ℓ rows and n columns called the share-generating matrix for Π. For all $i = 1, \ldots, \ell$, the i'th row of M, we let the function ρ defined the party labelling row i as $\rho(i)$. When we consider the column vector $v = (s, r_2, \ldots, r_n)$, where $s \in \mathbb{Z}_p$ is the secret to be shared, and $r_2, \ldots, r_n \in \mathbb{Z}_p$ are randomly chosen, then Mv is the vector of ℓ shares of the secret s according to Π. The share $(Mv)_i$ belongs to party $\rho(i)$.*

It is shown in [11] that every LSSS according to the above definition also enjoys the *linear reconstruction* property, defined as follows: Suppose that Π is an LSSS for the access structure $\mathbb{A}$. Let $S \in \mathbb{A}$ be any authorized set, and let $I \subset \{1, 2, \ldots, \ell\}$ be defined as $I = \{i : \rho(i) \in S\}$. Then, there exist constants $\{\omega_i \in \mathbb{Z}_p\}_{i \in I}$ such that, if $\{\lambda_i\}$ are valid shares of any secret s according to Π, then $\sum_{i \in I} \omega_i \lambda_i = s$. Furthermore, it is shown in [11] that these constants ω_i can be found in time polynomial in the size of the share-generating matrix M.

3 Definition of CP-ABE with SEM

We review the definition of CP-ABE with SEM in [25] here. Let S represent a set of attributes, and $\mathbb{A}$ an access structure. We defines $\mathbb{A}$ and S as the inputs to the encryption and key generation algorithm, and the function $f(S, \mathbb{A})$ outputs 1 iff the attribute set S satisfies the access structure $\mathbb{A}$, respectively.

Definition 3. *CP-ABE with SEM, $CP-ABE_{SEM}$ consists of five algorithms:*

$(PK, MSK) \leftarrow \mathsf{Setup}(\lambda, U)$. This algorithm takes security parameter λ and universe U as input. It outputs public parameters PK and master key MSK.

$CT \leftarrow \mathsf{Encrypt}(PK, m, \mathbb{A})$. This algorithm takes as input public parameters PK, a message m and an access structure $\mathbb{A}$. It outputs ciphertext CT.

$(SK_U, SK_M) \leftarrow \mathsf{KeyGen}(MSK, S)$. This algorithm takes as input master key MSK and an attribute set S. It outputs secret key SK_U and SEM's key SK_M.

$CT_M \leftarrow \mathsf{SEM\text{-}Decrypt}(SK_M, CT)$. The mediated decryption algorithm takes as input SEM's key SK_M for S and a ciphertext CT that was encrypted under $\mathbb{A}$. It outputs the partially decrypted ciphertext CT_M if $f(S, \mathbb{A}) = 1$. Otherwise, the error symbol $\perp$ is returned.

$M/\perp \leftarrow \mathsf{User\text{-}Decrypt}(SK_U, CT_M)$. The decrypt algorithm takes as input SK_U for S and CT_M that was originally encrypted under $\mathbb{A}$. It outputs the message m if $f(S, \mathbb{A}) = 1$ and (SK_U, SK_M) were created together. Otherwise, the error symbol $\perp$ is returned.

Some Terminologies. We define some terminologies and properties related to access structures here. Any monotonic (resp., non-monotonic) access structure $\mathbb{A}$ can be represented by a corresponding Boolean formula (resp., with negation), which we denote by $\phi(\mathbb{A})$, over variables in U. This is naturally defined in the sense that $f(S, \mathbb{A}) = 1$ holds iff the evaluation of $\phi(\mathbb{A})$ with the assignment that sets all variables in S to 1 and other variables outside S to 0 yields the value 1.

Consider the case where $\mathbb{A}$ is a monotonic access structure over U. If we denote a minimal representation of $\mathbb{A}$ by $\min(\mathbb{A}) = \{f(S, \mathbb{A}) = 1 |$ there exists no $B \in \mathbb{A}$such that $f(B, \mathbb{A}) = 1\}$. Then, it is straightforward to see that $\phi(\mathbb{A}) = \vee_{S' \in min(\mathbb{A})}(\wedge_{P \in S'} P)$.

For simplicity, we will use the access structure $\mathbb{A}$ and its corresponding Boolean formula $\phi(\mathbb{A})$ interchangeably when specifying a policy.

3.1 Fully-RCCA-CCA Security Model for CP-ABE with SEM

Consider a multi-party ABE with SEM system, where there are a lot of registered users and a mediator with real-time revocation list. An adversary $\mathcal{A}$ of time complexity of polynomial has the following capabilities.

1. $\mathcal{A}$ can corrupt users as his wish in the system to obtain their user secret keys SK_U and the corresponding SK_M from the mediator.

2. $\mathcal{A}$ can make SEM-Decrypt queries to get partially decrypted ciphertext CT_M.
3. $\mathcal{A}$ can make User-Decrypt queries to get plaintext m.

The goal of the adversary $\mathcal{A}$ is either of the two following outputs.

- A partially decrypted ciphertext CT_M for user key SK_U to decrypt although $\mathcal{A}$ has no knowledge of the corresponding SK_M. In particular, the user with SK_U has been revoked by SEM. So SEM will not help to partially decrypt ciphertext CT_M for the user. $\mathcal{A}$ tries to calculate CT_M without knowing SEM's SK_M. If it is the case, $\mathcal{A}$ can further decrypt CT_M successfully.
- A plaintext m decrypted from a partially decrypted ciphertext CT_M by SK_U. In particular, the user is valid and SEM helps to get CT_M. $\mathcal{A}$ tries to successfully decrypt CT_M to get m without knowing SK_U.

To clearly define $\mathcal{A}$'s capabilities and goal, we formally define two games.

Definition 4 [Security of a CP-ABE with SEM]. Let CP-ABE$_{SEM}$ = (Setup, Encrypt, KeyGen, SEM-Decrypt, User-Decrypt) be a CP-ABE with SEcurity Mediator scheme, $\mathcal{A}$ an adversary, $\lambda \in N$ a security parameter. We associate to CP-ABE$_{\mathbf{SEM}}$, $\mathcal{A}$ and λ an experiment $\mathbf{Exp}^{\mathbf{IND-RCCA-CCA}}_{CP-ABE_{SEM}}(\lambda)$ including the following two games.

In this definition, "RCCA-CCA" means security against users is RCCA while security against SEM is CCA. Now two Games are defined.

In Game-1, $\mathcal{A}$'s target is user.

Game-1

Setup. The challenger $\mathcal{C}$ runs Setup and gives PK to the adversary $\mathcal{A}$.

Phase 1. $\mathcal{C}$ initializes an empty table T_1, an empty set D_1 and an integer $j = 0$. Proceeding adaptively, $\mathcal{A}$ can repeatedly make any of these queries:

- Create(S). $\mathcal{C}$ sets $j := j + 1$. It runs KeyGen on S to obtain the pair (SK_U, SK_M) and stores in table T_1 the entry (j, S, SK_U, SK_M). It then returns to $\mathcal{A}$ SEM's key SK_M.

Note: Create can be repeatedly queried with the same input.

- Corrupt(i). If there exists an i^{th} entry in table T_1, then $\mathcal{C}$ obtains the entry (i, S, SK_U, SK_M) and sets $D_1 := D_1 \cup \{S\}$. It then returns to $\mathcal{A}$ user's secret key SK_U. If no such entry exists, then it returns $\perp$.
- Decrypt(i, CT_M). If there exists an i^{th} entry in table T_1, then $\mathcal{C}$ obtains the entry (i, S, SK_U, SK_M) and returns to $\mathcal{A}$ the output of the decryption algorithm on input (SK_U, CT_M). If no such entry exists, then it returns $\perp$.
- Challenge. $\mathcal{A}$ submits two equal-length messages m_0 and m_1. In addition, $\mathcal{A}$ gives a value $\mathbb{A}^*$ such that for all $S \in D_1$, $f(S, \mathbb{A}^*) \neq 1$. The challenger flips a random coin b, and encrypts m_b under $\mathbb{A}^*$. CT^* is given to $\mathcal{A}$.

Phase 2. Phase 1 is repeated with the restrictions that $\mathcal{A}$ cannot

- trivially obtain a secret key of user for the challenged ciphertext. That is, it cannot issue a Corrupt query that would result in a value S which satisfies $f(S, \mathbb{A}^*) = 1$ being added to D_1.

- issue a trivial decryption query. That is, Decrypt queries will be answered as in Phase 1, except that if the response would be either m_0 or m_1, then $\mathcal{C}$ responds with the special message *test* instead.

Guess. The adversary $\mathcal{A}$ outputs a guess b' of b.
The advantage of $\mathcal{A}$ in this game is $\mathsf{Adv}_{\text{Game-1}}(\mathcal{A}) = |Prob(b = b') - 1/2|$.

In Game-2, $\mathcal{A}$'s target is SEM.

Game-2

Setup. The challenger $\mathcal{C}$ runs Setup and gives PK to the adversary $\mathcal{A}$.
Phase 1. $\mathcal{C}$ initializes an empty table T_2, an empty set D_2 and an integer $j = 0$. Proceeding adaptively, $\mathcal{A}$ can repeatedly make any of these queries:

- Create(S). $\mathcal{C}$ sets $j := j + 1$. It runs KeyGen on S to obtain the pair (SK_U, SK_M) and stores in table T_2 the entry (j, S, SK_U, SK_M). It then returns to $\mathcal{A}$ user's secret key SK_U.

Note: Create can be repeatedly queried with the same input.

- Corrupt(i). If there exists an i^{th} entry in table T_2, then $\mathcal{C}$ obtains the entry (i, S, SK_U, SK_M) and sets $D_2 := D_2 \cup \{S\}$. It then returns to $\mathcal{A}$ SEM's key SK_M. If no such entry exists, then it returns $\perp$.

Challenge. $\mathcal{A}$ submits two equal-length messages m_0 and m_1. In addition, $\mathcal{A}$ gives a value $\mathbb{A}^*$ such that for all $S \in D_2, f(S, \mathbb{A}) = 1$. The challenger flips a random coin b, encrypts m_b under $\mathbb{A}^*$ and gives ciphertext CT^* to $\mathcal{A}$.
Phase 2. Phase 1 is repeated, except CT^* can not be queried.
Guess. The adversary $\mathcal{A}$ outputs a guess b' of b.
The advantage of $\mathcal{A}$ is defined as $\mathsf{Adv}_{\text{Game-2}}(\mathcal{A}) = |Prob(b = b') - 1/2|$.

The adversary $\mathcal{A}$ wins with advantage $\mathsf{Adv}_{\text{Game-1}}(\mathcal{A}) + \mathsf{Adv}_{\text{Game-2}}(\mathcal{A})$.
A CP-ABE with SEM scheme is RCCA-CCA-secure if $\mathsf{Adv}_{\text{Game-1}}(\mathcal{A}) + \mathsf{Adv}_{\text{Game-2}}(\mathcal{A}) < \epsilon$, where ϵ is negligible. □

4 Our Construction of CP-ABE with SEM

We now give our main construction of CP-ABE with SEM. Setup(λ, U). This algorithm takes as input the security parameter λ and the attribute universe description $U = \{0,1\}^*$. First we need to utilize a set W of dummy attributes, which is disjoint from U. A set of dummy attributes will then be associated to a verification key vk of a one-time signature scheme used in Encrypt algorithm. Set $W = \{P_{1,0}, P_{1,1}, P_{2,0}, P_{2,1}, \ldots, P_{\ell,0}, P_{\ell,1}\}$, where ℓ denotes the number of the rows in the LSSS matrix and $P_{i,j}$ are dummy attributes. Then choose the dummy attribute set $S_{vk} \subset W$ for all $vk \in \{0,1\}$ by setting $S_{vk} = \{P_{1,vk_1}, P_{2,vk_2}, \ldots, P_{\ell,vk_\ell}\}$. Then the algorithm chooses a bilinear group $\mathbb{G}$ of order $N = p_1p_2p_3$ (3 distinct primes). Use $\mathbb{G}_{p_i}$ to denote the subgroup of order p_i in $\mathbb{G}$. Then it chooses random exponents $\alpha, a \in \mathbb{Z}_N$, a random group element $g \in \mathbb{G}_{p_1}$ and two hash functions $H_1 : \{0,1\}^* \to \mathbb{Z}_p$ and $H_2 : \{0,1\}^* \to \{0,1\}^k$. For each attribute $i \in \{U \cup W\}$, it chooses a random value $s_i \in \mathbb{Z}_N$.

The public parameters PK are $N, g, g^a, e(g,g)^\alpha, T_i = g^{s_i} \forall i, H_1, H_2$ and a generator X_3 of $\mathbb{G}_{p_3}$. The master secret key MSK is g^α.

KeyGen(MSK, S, PK). This algorithm takes as input the master secret key MSK, public key PK and an attribute set S, where $S \subset U$. It outputs the user's key SK_U and SEM's key $SK_M^{S'}$ for the attribute set S' where $S' = S \cup W$. The algorithm chooses a random $t' \in \mathbb{Z}_N$, and random elements $R_0, R_0', R_i \in \mathbb{G}_{p_3}$. Let $SK' = (PK, K' = g^\alpha g^{at'} R_0, L' = g^{t'} R_0', \{K_x' = T_i^{t'} R_i\}_{x \in S'})$. Then choose $z \in_R \mathbb{Z}_p^*$ as user's secret key SK_U. Let $t = t'/z$ and set SEM's key as

$$SK_M^{S'} = (PK, K = K'^{1/z} = g^{\alpha/z} g^{at'/z} = g^{\alpha/z} g^{at}, L = L'^{1/z} = g^{t'/z} = g^t, \{K_x\}_{x \in S'} = \{K_x'^{1/z}\}_{x \in S'}).$$

Encrypt$(PK, \mathbb{A} = (M, \rho), m \in \{0,1\}^k)$. It takes as input an LSSS access structure $\mathbb{A} = (M, \rho)$, the public parameters PK and a message m to encrypt, where M is an $\ell \times n$ LSSS matrix and the function ρ associates each row M_i to attribute $\rho(i)$. Let the dummy policy $\mathbb{A}' = \mathbb{A} \wedge (\wedge_{P \in S_{vk}} P)$. The algorithm outputs the ciphertext encrypted under policy $\mathbb{A}' = (M', \rho')$, where M' is an $\ell' \times n'$ LSSS matrix and the function ρ' associates each row $M_n' i$ to attribute $\rho'(i)$. The algorithm chooses a vector $\boldsymbol{v} \in_R \mathbb{Z}_p^{n'}$, denoted $\boldsymbol{v} = (s, v_2, \ldots, v_{n'})$. For each row of M', i.e., the vector $\boldsymbol{M_i'}$, calculate $\lambda_i = \boldsymbol{v} \cdot \boldsymbol{M_i'}$. Then choose $r_i \in_R \mathbb{Z}_p$ for $i = 1, \cdots, \ell'$. Then it selects a random $R \in \mathbb{G}_T$ and computes $s = H_1(R, m)$ and $r = H_2(R)$. An attribute parameter can be derived into an element in the group $\mathbb{G}$ by a function $F : \{0,1\}^* \to \mathbb{G}$. The ciphertext CT is shown as

$$(C = Re(g,g)^{\alpha s}, C' = g^s, C'' = m \oplus r, C_i = g^{a\lambda_i} \cdot F(\rho'(i))^{-r_i}, D_i = g^{r_i} \ \forall i),$$

and CT implicitly contains the access structure $\mathbb{A}' = (M', \rho')$.

Let $\Sigma = (\mathcal{G}, \mathcal{S}, \mathcal{V})$ be a one-time signature scheme. The algorithm creates a one-time signature key pair by running $\mathcal{G} \to (vk, sk)$. It then runs $\mathcal{S}(sk, CT) \to \sigma$. And the algorithm outputs $CT' = (vk, CT, \sigma)$.

SEM-Decrypt$(PK, CT', SK_M^{S'})$. It takes as input a ciphertext CT' for a linear access structure $\mathbb{A}' = (M', \rho')$, PK and SEM's key $SK_M^{S'} = (PK, K, L, \{K_x\}_{x \in S'})$ for a set S', where $S' = S \cup W$. It parses the ciphertext CT' as (vk, CT, σ). If $\mathcal{V}(vk, CT, \sigma) = 0$, then it outputs $\perp$.

Otherwise, the mediator can get $SK_M^{S \cup S_{vk}}$ as $(PK, K^{S \cup S_{vk}}, L^{S \cup S_{vk}}, \{K_x^{S \cup S_{vk}}\}_{x \in S})$ for the set $S \cup S_{vk}$ within the method **Delegate** described in [13] as follows. Since S_{vk} is a subset of W, $S \cup S_{vk}$ is a subset of S'. The algorithm random chooses $u \in \mathbb{Z}_N$ and random elements $R_0, R_0', R_i \in G_{p_3}$, and computes

$(PK, K^{S \cup S_{vk}} = K g^{au} R_0, L^{S \cup S_{vk}} = L g^u R_0', \{K_x^{S \cup S_{vk}}\}_{x \in S} = K_i T_i^u R_i, \forall i \in S \cup S_{vk})$ as $SK_M^{S \cup S_{vk}}$.

Let $I \subset \{1, 2, \cdots, \ell\}$ be defined as $I = \{i : \rho(i) \in S \cup S_{vk}\}$. Then, let $\{\omega_i \in \mathbb{Z}_p\}_{i \in I}$ be a set of constants such that if $\{\lambda_i\}$ are valid shares of any secret s according to M, then $\sum_{i \in I} \omega_i \lambda_i = s$. Then SEM computes

$$e(C', K^{S\cup S_{vk}})/\left(e(\prod_{i\in I} C_i^{\omega_i}, L^{S\cup S_{vk}}) \cdot \prod_{i\in I}(e(D_i^{\omega_i}, K_{\rho(i)}^{S\cup S_{vk}}))\right)$$
$$= e(g^s, g^{\alpha/z} g^{at} g^{au})/\left(e(\prod_{i\in I} g^{a\lambda_i\omega_i}, g^t g^u) \cdot \prod_{i\in I} e(T_{\rho(i)}^{-r_i}, g^t g^u) \prod_{i\in I}(e(g^{r_i\omega_i}, T_{\rho(i)}^{tu}))\right)$$
$$= e(g^s, g^{\alpha/z}) e(g^s, g^{at}) e(g^s, g^{au})/\left(e(g^{as}, g^t g^u) \prod_{i\in I} e(T_{\rho(i)}^{-r_i}, g^t g^u) \prod_{i\in I} e(g^{r_i}, T_{\rho(i)}^{tu})\right)$$
$$= e(g,g)^{\alpha s/z},$$

and sends the partially decrypted ciphertext $CT_M = (C, C'', e(g,g)^{\alpha s/z})$ to user. User-Decrypt(CT_M, SK_U). It takes as input a partially decrypted ciphertext CT_M for a linear access structure (M, ρ), user's secret key SK_U for a set S. It parses CT_M as (CT_0, CT_1, CT_2), and computes $R = CT_0/CT_2^z$, $m = CT_1 \oplus H_2(R)$, and $s = H_1(R, m)$. If $CT_0 = R \cdot e(g,g)^{\alpha s}$ and $CT_2 = e(g,g)^{\alpha s/z}$, it outputs m; otherwise, it outputs the error symbol $\perp$.

In our construction, if a user has appropriate attribute set S to match the access structure $\mathbb{A}$ on the ciphertext CT, i.e. $f(S, \mathbb{A}) = 1$, the user can decrypt the partially decrypted ciphertext CT_M correctly and get the plaintext.

User Revocation. In our construction, Security Mediator holds a revocation list (RL), which records the revoked users. The mediator can add a user into the revocation list or delete a user from RL. When execute SEM-Decrypt algorithm, SEM will check the revocation list first. If a user is revoked, Security Mediator won't pass the partially decrypted ciphertext CT_M to the user, and the revoked user cannot finish the User-Decrypt algorithm to get the plaintext.

4.1 Security Proof

Theorem 1. *Let* Π *be a Waters' scheme in [11] and* Σ' *is our* $CP-ABE_{SEM}$ *scheme. If* Π *is a CPA-secure CP-ABE scheme, then our scheme* $CP-ABE_{SEM}$ *is fully-RCCA-secure.*

Theorem 2. *Let* Π *be a Waters' scheme in [11] and* Σ *is a one-time signature,* Σ' *is our* $CP-ABE_{SEM}$ *scheme. If* Π *is a* $(\tau, \epsilon_{ABE}, q)$*CPA-secure CP-ABE scheme,* Σ *is* (τ, ϵ_{OTS}) *secure, then our scheme* $CP-ABE_{SEM}$ *is*$(\tau - o(\tau), \epsilon_{ABE} + \epsilon_{OTS}, q_D, q_E)$ *fully-CCA-secure where* $q \geq q_D + q_E$.

Proof 1. Suppose there exists a polynomial-time adversary $\mathcal{A}$, who can break our scheme in our new fully RCCA-security model with non-negligible advantage ϵ, Then we can build a simulator $\mathcal{B}$ for $\mathcal{A}$ who can break Waters' scheme in [11] in the selective CPA-secure model with advantage ϵ minus a negligible amount. However, in [11], Waters' scheme has been proven secure under the decisional parallel BDHE assumption. That is a contradiction.

Setup. $\mathcal{B}$ obtains the public parameter $PK = (g, e(g,g)^{\alpha}, g^a, F)$, where F is a hash function from Waters' scheme. $\mathcal{B}$ initializes the empty tables T_1, T_{H_1}, T_{H_2}, an empty set D, and an integer $j = 0$. It sends PK to $\mathcal{A}$ as the public parameter.

Phase 1. $\mathcal{B}$ answers $\mathcal{A}$'s queries as follows.

- Random Oracle Hash $H_1(R, m)$. $\mathcal{B}$ checks the table T_{H_1} first. If there exists an entry (R, m, s) in T_{H_1}, returns s to $\mathcal{A}$. Otherwise, $\mathcal{B}$ chooses $s \in_R \mathbb{Z}_p$, records (R, m, s) in T_{H_1} and returns s to $\mathcal{A}$.
- Random Oracle Hash $H_2(R)$. $\mathcal{B}$ checks the table T_{H_2} first. If there exists an entry (R, r) in T_{H_2}, returns r to $\mathcal{A}$. Otherwise, $\mathcal{B}$ chooses $r \in_R \mathbb{Z}_p$, records (R, r) in T_{H_2} and returns r to $\mathcal{A}$.
- $\mathsf{Create}(S)$: $\mathcal{B}$ sets $j := j + 1$ and proceeds one of two ways.

– If S satisfies $\mathbb{A}^* = (M^*, \rho^*)$, i.e., $f(S, \mathbb{A}^*) = 1$, $\mathcal{B}$ chooses a fake key pair (SK_U, SK_M): choose $d \in_R \mathbb{Z}_p$ and run $\mathsf{KeyGen}(g^d, S, PK)$ to obtain SK'

$$SK' = (PK, K' = g^d g^{at'}, L' = g^{t'}, \left\{K'_x = F(x)^{t'}\right\}_{x \in S}).$$

Set $SK_M = SK'$. Let $d = \alpha/z$, replace d by α/z and we have $SK_M =$

$$(PK, g^{\alpha/z} g^{at'}, L' = g^{t'}, \left\{F(x)^{t'}\right\}_{x \in S}) =$$
$$(PK, K = K'^{1/z} = (g^{\alpha} g^{at''})^{1/z}, L = L'^{1/z} =$$
$$g^{t''/z}, \left\{K_x = K_x'^{1/z} = (F(x)^{t''})^{1/z}\right\}_{x \in S}).$$

By the replacement, SK_M is properly distributed.

– Otherwise, $\mathcal{B}$ calls Waters' key generation oracle on S to obtain $SK' = (PK, K', L', \{K'_x\}_{x \in S})$. Then, it chooses $z \in_R \mathbb{Z}_p$ and sets

$$SK_M = (PK, K = K'^{1/z}, L = L'_{1/z}, \left\{K_x = K_x'^{1/z}\right\}_{x \in S}), SK_U = z.$$

Finally, $\mathcal{B}$ stores (j, S, SK_U, SK_M) in the table T and returns SK_M to $\mathcal{A}$.

- $\mathsf{Corrupt}(i)$: If no such entry exists (i.e., $i > j_{max}$), or if S in the i^{th} entry (i, S, SK_U, SK_M) satisfies $\mathbb{A}^*$, $\mathcal{B}$ returns $\perp$. Otherwise, $\mathcal{B}$ returns SK_U to $\mathcal{A}$.
- $\mathsf{Decrypt}(i, CT_M)$. Let $CT_M = (CT_0, CT_1, CT_2)$ be associated with an access structure $\mathbb{A} = (M, \rho)$. Let j_{max} denote the currently maximum j of the table T.

If $i > j_{max}$, there is no satisfactory entry (i, S, SK_U, SK_M) exists. Then returns $\perp$. If there exists the i^{th} entry (i, S, SK_U, SK_M) in T while $f(S, \mathbb{A}) \neq 1$, $\mathcal{B}$ returns $\perp$, too. Then $\mathcal{B}$ proceeds one of the following two ways.

– If the i^{th} entry (i, S, SK_U, SK_M) does not satisfy the challenge structure $\mathbb{A}^* = (M^*, \rho^*)$, it proceeds as follows.

1. Compute $R = C_0 / C_2^z$.

2. Obtain the records (R, m_x, s_x) from T_{H_1}. If no such record exists, return $\perp$. If there exists indices $x_1 \neq x_2$ such that (R, m_{x_1}, s_{x_1}) and (R, m_{x_2}, s_{x_2}) are in T_{H_1}, and $m_{x_1} \neq m_{x_2}$ while $s_{x_1} = s_{x_2}$, $\mathcal{B}$ aborts. As s_{x_1} and s_{x_2} are randomly chosen in $\mathbb{Z}_p$, the probability $\mathcal{B}$ aborts is $(1 - p!/((p - q_{H_1})! \cdot p^{q_{H_1}}))$.
3. Obtain the records (R, r) from T_{H_2}. If no such record exists, return $\perp$.
4. Test if $CT_0 = R(e(g,g)^{\alpha})^{s_x}, CT_1 = m \oplus r, CT_2 = e(g,g)^{\alpha s_i/z}$, for each i in the records (R, m_x, s_x) in step 2.

If there is an x passes the above test, return m_x to $\mathcal{A}$. Otherwise return $\perp$.
- If the i^{th} entry (i, S, SK_U, SK_M) satisfies (M^*, ρ^*), it proceeds as follows.

1. Compute $\beta = C_2^{1/d}$.
2. For each record (R_x, m_x, s_x) in table T_{H_1}, test if $\beta = e(g,g)^{s_x}$.
3. If no match is found, $\mathcal{B}$ returns $\perp$.
4. If more than one match are found, $\mathcal{B}$ aborts the simulation.
5. Otherwise, let (R, m, s) be the sole match. Obtain the record (R, r) in table T_{H_2}. If it does not exist, $\mathcal{B}$ returns $\perp$.
6. Test if $CT_0 = R \cdot e(g,g)^{\alpha s}$, $CT_1 = m \oplus r$ and $CT_2 = e(g,g)^{ds}$. If all tests pass, output m; else, output $\perp$.

Challenge. $\mathcal{A}$ submits two messages $(m_0^*, m_1^*) \in \{0,1\}^{2k}$. $\mathcal{B}$ acts as follows.

1. $\mathcal{B}$ chooses random $(R_0, R_1) \in \mathbb{G}_T^2$ and passes them onto Waters' challenger to obtain ciphertext $CT_b = (C, C', \{C_i, D_i\}_{i \in \{1, \cdots, \ell\}})$ under (M^*, ρ^*), $b \in \{0,1\}$.
2. $\mathcal{B}$ chooses a random value $C'' \in \{0,1\}^k$.
3. $\mathcal{B}$ sends the ciphertext $CT'' = (C, C', C'', \{C_i, D_i\}_{i \in [1, \cdots, \ell]})$ to $\mathcal{A}$.

Phase 2. $\mathcal{B}$ continues to answer queries as in Phase 1, except that if the response to a Decrypt query would be either m_0^* or m_1^*, then $\mathcal{B}$ answers **test**.

Guess. Eventually, $\mathcal{A}$ must either output a bit or abort, either way $\mathcal{B}$ ignores it. R_0 and R_1 are never revealed to $\mathcal{A}$ except in the challenge ciphertext. A necessary condition for $\mathcal{A}$ to win is to query the hash of the value it obtained from the challenge ciphertext. Since the advantage for $\mathcal{A}$ to break the scheme is ϵ, the probability for $\mathcal{A}$ to query to either or both R_0 and R_1 should be at least ϵ. (If $\mathcal{A}$ does not make the query, it can only win by random guessing.) Next, $\mathcal{B}$ searches through tables T_{H_1} and T_{H_2} to see if R_0 or R_1 appears as the first value in any entry (i.e., $\mathcal{A}$ once issued a query of the form $H_1(R_b)$or $H_2(R_b)$.)

$\mathcal{B}$'s advantage in Game 1 is obviously negligible, and Theorem 1 is RCCA secure within this negligible advantage. The proof of Theorem 1 is complete. □

Proof 2. Assume we are given an adversary $\mathcal{A}$ which breaks CCA-security of our scheme Σ' with running time τ, advantage ϵ, q_E key-extraction queries, and, q_D decryption queries. We use $\mathcal{A}$ to construct another adversary $\mathcal{B}$ which breaks CPA-security of the ABE scheme Σ. Describe the game 2 as follows:

Setup. The challenger runs $\mathsf{Setup}(\lambda, U \cup W) \rightarrow (PK, MSK)$. Then $\mathcal{B}$ is given PK and gives it to $\mathcal{A}$. $\mathcal{B}$ also runs $\mathcal{G}(\lambda) \rightarrow (vk^*, sk^*)$.

Phase 1. $\mathcal{A}$ may adaptively make queries of the following types:

- Key-extraction query. When $\mathcal{A}$ submits S, $\mathcal{B}$ submits $S \cup W$ to the challenger. $\mathcal{B}$ is given $SK^{S\cup W}$ for $S \cup W$ and chooses a random $z \in \mathbb{Z}_N$. $\mathcal{B}$ calculates $SK_M^{S\cup W} = (SK^{S\cup W})^{1/z}$ and $SK_U = z$, then gives them to $\mathcal{A}$.
- Decryption query. When $\mathcal{A}$ submits (CT', S) such that $CT' = (vk, CT, \sigma)$, $\mathcal{B}$ first checks whether $\mathcal{V}(vk, CT, \sigma)$ holds. If it does not hold, then $\mathcal{B}$ returns $\perp$. If it holds and $vk^* = vk$, then $\mathcal{B}$ aborts. Otherwise, $\mathcal{B}$ submits $S \cup S_{vk}$ to the challenger and is given $SK_{S\cup S_{vk}}$. Then $\mathcal{B}$ rerandomizes it by $SK^{S\cup S_{vk}} \leftarrow \mathsf{Delegate}(PK, SK^{S\cup S_{vk}}, S \cup S_{vk}, S \cup S_{vk})$ and calculates $SK_M^{S\cup S_{vk}}$ by randomly choose $z \in \mathbb{Z}_N$. It returns output of $\mathsf{SEM\text{-}Decrypt}(PK, CT, SK_M^{S\cup S_{vk}})$ to $\mathcal{A}$.

Challenge. $\mathcal{A}$ declares two equal length messages m_0^*, m_1^* and $\mathbb{A}^*$. Then $\mathcal{B}$ declares the same messages m_0^*, m_1^* and $\mathbb{A}'^*$ for the challenger, where $\mathbb{A}'^*$ is an access structure such that $\psi(\mathbb{A}'^*) = \psi(\mathbb{A}^*) \wedge (\wedge_{P\in S_{vk}} P)$. The challenger flips a random coin $b \in \{0, 1\}$, runs $\mathsf{Encrypt}(PK, M_\beta, \psi(\mathbb{A}'^*) \rightarrow CT^*$ and gives CT^* to $\mathcal{B}$. Then $\mathcal{B}$ runs $\mathcal{S}(sk^*, CT^*) \rightarrow \sigma^*$ and gives $CT'^* = (VK^*, CT^*, \sigma^*)$ to $\mathcal{A}$ as challenge ciphertext. $\mathcal{B}$ also choose a random $z \in \mathbb{Z}_N$ as SK_U and gives it to $\mathcal{A}$.

Phase 2. $\mathcal{B}$ answers $\mathcal{A}$'s query where $f(S, \mathbb{A}^*) \neq 1, CT \neq CT^*$ as in Phase 1.

Guess. Finally, $\mathcal{A}$ outputs a guess b' for b. Then $\mathcal{B}$ outputs b' as its guess.

Let Win denote the event that $\mathcal{A}$ guess b correctly, Abort denote the event that $\mathcal{B}$ aborts. If Abort does not occur, $\mathcal{B}$'s simulation is perfect. So, $\mathcal{B}$'s advantage for guessing β is estimated as $Pr[\mathcal{B} \text{ correctly guess } \beta] - \frac{1}{2} = Pr[\mathsf{Win}|\overline{\mathsf{Abort}}]Pr[\overline{\mathsf{Abort}}] - \frac{1}{2} \geq \epsilon - Pr[\overline{\mathsf{Abort}}]$. Since $Pr[\overline{\mathsf{Abort}}] \leq \epsilon_{OTS}$ holds due to the unforgeability of the one-time-signature, Theorem 2 holds. Thus the proof is completed. □

According to the proofs of Theorems 1 and 2 above, our scheme $CP-ABE_{SEM}$ is a fully-RCCA-CCA-secure CP-ABE with SEM scheme.

5 Conclusion and Discussion

User revocation and decryption outsourcing are two issues for Attribute-Based Encryption scheme apart from practice. To solve them, we propose a fully-ReplayableCCA-CCA-secure Ciphertext-Policy-ABE with SEcurity Mediator, CP-ABE with SEM for short. It introduces SEM for checking the validity of user immediately with a real-time revocation list, and partially decrypting the ciphertext if the user is unrevoked. One interesting future work is to further reduce the computation overload by using prime-order cyclic group without composite-order one. Another direction is to construct a fully-CCA-secure ABE with SEM.

Acknowledgments. The paper is funded by the National Natural Science Foundation of China under Grants 61402136 and 61240011, Shenzhen Development and Reform Commission [2012]720, Shenzhen Development and Reform Commission [2012]900, Shenzhen Basic Research JC201104210032A and JC201005260112A, and the Seed Funding Programme for Basic Research, HKU 201311159040.

References

1. Sahai, A., Waters, B.: Fuzzy identity-based encryption. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 457–473. Springer, Heidelberg (2005)
2. Goyal, V., Pandey, O., Sahai, A., Waters, B.: Attribute-based encryption for fine-grained access control of encrypted data. In: 13th ACM Conference on Computer and Communications Security (CCS 2006), pp. 89–98 (2006)
3. Canetti, R., Halevi, S., Katz, J.: Chosen-ciphertext security from identity-based encryption. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 207–222. Springer, Heidelberg (2004)
4. Bethencourt, J., Sahai, A., Waters, B.: Ciphertext-policy attribute-based encryption. In: IEEE Symposium on Security and Privacy (SP 2007), pp. 321–334 (2007)
5. Cheung, L., Newport, C.: Provably secure ciphertext policy ABE. In: 14th ACM conference on Computer and communications security (CCS 2007), pp. 456–465 (2007)
6. Ostrovsky, R., Sahai, A., Waters, B.: Attribute-based encryption with non-monotonic access structures. In: 14th ACM Conference on Computer and Communications Security (CCS 2007), pp. 195–203 (2007)
7. Boldyreva, A., Goyal, V., Kumar, V.: Identity-based Encryption with Efficient Revocation. In: ACM conference on Computer and communications security (2008)
8. 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)
9. Liang, X., Lu, R., Lin, X., Shen, X.: Ciphertext policy attribute-based encryption with efficient revocation. Technical report, University of Waterloo (2010)
10. Qian, J., Dong, X.: Fully secure revocable attribute-based encryption. J. Shanghai Jiaotong Univ. (Sci.) **16**, 490–496 (2011)
11. Waters, B.: Ciphertext-policy attribute-based encryption: an expressive, efficient, and provably secure realization. In: Public Key Cryptography (PKC 2011), pp. 53–70 (2011)
12. 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)
13. Yamada, S., Attrapadung, N., Hanaoka, G., Kunihiro, N.: Generic constructions for chosen-ciphertext secure attribute based encryption. In: Catalano, D., Fazio, N., Gennaro, R., Nicolosi, A. (eds.) PKC 2011. LNCS, vol. 6571, pp. 71–89. Springer, Heidelberg (2011)
14. 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)
15. Okamoto, T., Takashima, K.: Fully secure functional encryption with general relations from the decisional linear assumption. In: Rabin, T. (ed.) CRYPTO 2010. LNCS, vol. 6223, pp. 191–208. Springer, Heidelberg (2010)
16. Green, M., Hohenberger, S., Waters, B.: Outsourcing the Decryption of ABE Ciphertexts. In: USENIX Security Symposium (2011)
17. Lynn, B.: The Stanford Pairing Based Crypto Library. http://crypto.stanford.edu/pbc
18. Hur, J., Noh, D.K.: Attribute-based access control with efficient revocation in data outsourcing systems. IEEE Trans. Parallel Distrib. Syst. **22**(7), 1214–1221 (2011)

19. Yu, S., Wang, C., Ren, K., Lou, W.: Attribute based data sharing with attribute revocation. In: Proceedings of the 5th ACM Symposium on Information. Computer and Communications Security (AsiaCCS 2010), pp. 261–270 (2010)
20. Boneh, D., Ding, X., Tsudik, G.: Fine-grained control of security capabilities. ACM Trans. Internet Technol. (TOIT) **4**(1), 60–82 (2004)
21. Chow, S.S.M., Boyd, C., González Nieto, J.M.: Security-mediated certificateless cryptography. In: Yung, M., Dodis, Y., Kiayias, A., Malkin, T. (eds.) PKC 2006. LNCS, vol. 3958, pp. 508–524. Springer, Heidelberg (2006)
22. Gennaro, R., Gentry, C., Parno, B.: Non-interactive verifiable computing: outsourcing computation to untrusted workers. In: Rabin, T. (ed.) CRYPTO 2010. LNCS, vol. 6223, pp. 465–482. Springer, Heidelberg (2010)
23. Chung, K.-M., Kalai, Y., Vadhan, S.: Improved delegation of computation using fully homomorphic encryption. In: Rabin, T. (ed.) CRYPTO 2010. LNCS, vol. 6223, pp. 483–501. Springer, Heidelberg (2010)
24. Canetti, R., Krawczyk, H., Nielsen, J.B.: Relaxing chosen-ciphertext security. In: Boneh, D. (ed.) CRYPTO 2003. LNCS, vol. 2729, pp. 565–582. Springer, Heidelberg (2003)
25. Ibraimi, L., Petkovic, M., Nikova, S., Hartel, P., Jonker, W.: Mediated ciphertext-policy attribute-based encryption and its application. In: Youm, H.Y., Yung, M. (eds.) WISA 2009. LNCS, vol. 5932, pp. 309–323. Springer, Heidelberg (2009)
26. Pirretti, M., Traynor, P., McDaniel, P., Waters, B.: Secure attribute-based systems. In: ACM Conference on Computer and Communications Security, pp. 99–112 (2006)
27. Pirretti, M., Traynor, P., McDaniel, P., Waters, B.: Secure attribute-based systems. J. Comput. Secur. **18**(5), 799–837 (2010)
28. 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)
29. Yang, K., Jia, X., Ren, K., Huang, L.: Enabling efficient access control with dynamic policy updating for big data in the cloud. In: Proceedings IEEE on INFOCOM 2014, pp. 2013–2021 (2014)