IEICE TRANS. INF. & SYST., VOL.E100–D, NO.10 OCTOBER 2017

PAPER Special Section on Advanced Log Processing and Office Information Systems

Attribute Revocable Multi-Authority Attribute-Based Encryption with Forward Secrecy for Cloud Storage

Kenta NOMURA[†*a)], ***Nonmember***, **Masami MOHRI**[††], **Yoshiaki SHIRAISHI**[†], ***and*** **Masakatu MORII**[†], ***Senior Members***

SUMMARY Internet of Things (IoT) has been widely applied in various fields. IoT data can also be put to cloud, but there are still concerns regarding security and privacy. Ciphertext-Policy Attribute-Based Encryption (CP-ABE) is attracted attention in cloud storage as a suitable encryption scheme for confidential data share and transmission. In CP-ABE, the secret key of a user is associated with a set of attributes; when attributes satisfy the access structure, the ciphertext is able to be decrypted. It is necessary that multiple authorities issue and manage secret keys independently. Authorities that generate the secret key can be regarded as managing the attributes of a user in CP-ABE. CP-ABE schemes that have multiple authorities have been proposed. The other hand, it should consider that a user's operation at the terminals is not necessary when a user drop an attribute and key is updated and the design of the communication system is a simple. In this paper, we propose CP-ABE scheme that have multiple key authorities and can revoke attribute immediately with no updating user's secret key for attribute revocation. In addition, the length of ciphertext is fixed. The proposed scheme is IND-CPA secure in DBDH assumption under the standard model. We compare the proposed scheme and the other CP-ABE schemes and show that the proposed scheme is more suitable for cloud storage.

key words: *ciphertext-policy attribute-based encryption, multiple key authorities, attribute revocation, forward secrecy*

1. Introduction

With the development of wireless sensor networks, global positioning system, and other related techniques, Internet of Things (IoT) has been widely applied in many applications successfully and plays an important role in intelligent transportation system (ITS), smart grid and so on. IoT datasets can be very large when data is generated over a certain amount of time. The amount of data which are created and copied will reach 44 zettabytes, or 44 trillion gigabytes [1]. IoT data can also be put to cloud for processing [2]. There are still strong concerns regarding data security and user privacy, the researches in this area are carried [3]–[6]. To guarantee security of confidential data is one of the major challenges when data owners store confidential data on external cloud server. Granting access rights to certain users and forbidding other users to the data, which is called access control, ensure confidentiality of data. One way to achieve access control to attach a list of all valid users to the data. However, in cloud scenario, such lists can be extremely long and often dynamic, which make handling such lists extremely difficult. Another way to prevent invalid users getting data is encrypting data by using public keys of valid users, so that only they are able to decrypt data using their secret keys. However, the same data must be encrypted several times individually for each user, which may result in huge storage costs and calculation costs. Hence Attribute-Based Encryption (ABE) has been attracting attention as the access control method in cloud [7]–[16].

Sahai *et al.* [17] have proposed ABE as the method that extends the Identity-based encryption [18] which distinguish individuals on strings. ABE comes in two types called key-policy ABE (KP-ABE) [19] and ciphertext-policy ABE (CP-ABE) [20]. In KP-ABE, the encryptor labels each ciphertext with a set of attributes. Key authorities generate the secret key which is associated with an access structure that specifies which type of ciphertext the key can decrypt. The user who can decrypt the ciphertext is controlled by not the encryptor but key authorities. An application example include encryption of log in forensic analysis and broadcast encryption [19]. On the other hand, in CP-ABE, the secret key of a user is associated with a set of attributes; when attributes satisfy the access structure, the ciphertext is able to be decrypted. The access control scheme for cloud storage by exploiting KP-ABE has been proposed [10]. However the disadvantage of KP-ABE is that the access structure is built into a user's secret key, so data owner cannot choose who can decrypt the data except choosing a set of attributes which can describe this data. Thus, CP-ABE is more appropriate to cloud storage than KP-ABE. CP-ABE schemes for cloud storage have proposed [7]–[9], [11]–[16]. Initially, in ABE schemes, a single authority generate the whole secret keys of users [17], [20]–[24]. Chase *et al.* [25] and Lewko *et al.* [26] have proposed ABE schemes which multiple authorities generate the secret key in. These schemes use the concepts of a trusted central authority (CA) and global identifiers. However, the CA has the power to decrypt every ciphertext, so the multiple authority ABE schemes without a CA have proposed [27], [28]. If a single authority manages the master secret key, the enemy may get all the master secret key when the enemy attacks the authority. In other schemes [29], [30], multiple authorities generate the secret key while hiding each of the master secret keys. In IoT environment, multiple services and applications may

Manuscript received November 14, 2016.
Manuscript revised March 31, 2017.
Manuscript publicized July 21, 2017.
[†]The authors are with Kobe University, Kobe-shi, 657–8501 Japan.
[††]The author is with Gifu University, Gifu-shi, 501–1193 Japan.
[*]Presently, with PwC Cyber Services LLC.
a) E-mail: kenta.nomura@pwc.com
DOI: 10.1587/transinf.2016OFP0004

use the common cloud server. User's attributes managed independently by each service and application. Therefore multiple authorities must generate the secret key.

The issue of key revocation is one of the significant challenges. Bethencourt *et al*. [20] and Boldyreva *et al*. [31] have firstly suggested key revocation mechanisms in CP-ABE and KP-ABE, respectively. Their mechanisms are to append to each attribute an expiration date and distribute new keys to valid users after the expiration. These periodic attribute revocable ABE schemes [20], [28], [31]–[33] do not meet forward secrecy. Forward secrecy means that any user who drops an attribute cannot decrypt ciphertext after he drops the attribute. In the periodic attribute revocable ABE schemes, if an authority revokes a specific user at any time, that user can decrypt ciphertext until the expiration. The immediate revocable schemes have proposed. The immediate system-level user revocation method [16], [21], [34], [35] and the immediate attribute-level key revocation method [12], [15], [30], [36], [37] have proposed. The concept of negative attribute enable to revoke the user at the system level. Each user has an attribute that represents his own identifier (ID). Then, the idea for revoking users is to attach a negative constraint to the ciphertext's access structure which include the IDs of the revoked users. In this method, it is difficult to re-issue the secret key. It is desirable to revoke the user by the specific attribute unit.

Users have not only the static attributes whose value remain unchanged but also the dynamic attributes which need to be updated. For example, in vehicular ad hoc networks, vehicles have the dynamic attributes which represent the position information [38]. The *dynamic* attribute does not mean that attribute fields are newly added or deleted but mean that some specific attribute field value can be changed. CP-ABE schemes which have capability of revoking some current attributes have been proposed [12], [15], [29]–[32], [37]. In these schemes, the secret keys are needed to update when an attribute revocation happens. Users cannot update their secret key when the key is embedded in IoT devices or they cannot access the key authorities which manage attributes.

The ABE schemes can be classified as the length of ciphertext is fixed or variable. In CP-ABE schemes in which the length of ciphertext is variable, the length of ciphertext depends on the number of attributes which are included in an access structure. Fixed length ciphertexts make processes of accumulating or transmitting data simpler than variable length ciphertexts in IoT or cloud environment. For example, if the length of ciphertext is constant, it would be easy to estimate desired capacity of storage or channel for storing or sharing data.

This paper suggests the following five requirements when the CP-ABE schemes are applied for cloud storage.

Requirement 1: Multiple key authorities independently manage attributes

Requirement 2: The key revocation is flexible

Requirement 3: That scheme meets security requirements
1: collusion resistance
2: data confidentiality
3: backward and forward secrecy
and are proved security

Requirement 4: Attribute revocation does not change the secret key

Requirement 5: The length of ciphertext is fixed

We propose the CP-ABE scheme which meets the above requirements and show that our scheme is more suitable for cloud storage than the other CP-ABE schemes by comparing these.

2. Preliminaries

2.1 Bilinear Maps

Let $\mathbb{G}$ and $\mathbb{G}_T$ be two multiplicative cyclic groups of prime order p. Let $P \in \mathbb{G}$ be a generator of $\mathbb{G}$ and e be a bilinear map, e: $\mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$. For all $a,\ b \in \mathbb{Z}_p$, the bilinear map e has the following properties:

1) Bilinearity: $e(aP, bP) = e(P, P)^{ab}$
2) Non-degeneracy: $e(P, P) \neq 1$

We say that $\mathbb{G}$ is a bilinear group if the group operation in $\mathbb{G}$ and bilinear map e: $\mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ are both efficiently computable. Notice that the map e is symmetric since $e(aP, bP) = e(P, P)^{ab} = e(bP, aP)$. Then, in this paper, we define exponentiation as $nP := P^n$.

2.2 Decisional Bilinear Diffie-Hellman (DBDH) Assumption

Let $a,\ b,\ c,\ z \in \mathbb{Z}_p$ be chosen at random and P be a generator of $\mathbb{G}$. DBDH assumption is that no probabilistic polynomial time adversary $\mathcal{A}$ is able to distinguish the tuples $\langle aP, bP, cP, e(P, P)^{abc}\rangle$ and $\langle aP, bP, cP, e(P, P)^{z}\rangle$ with non-negligible advantage, where the advantage of $\mathcal{A}$ is defined as

$$\mathrm{Adv}_{\mathcal{A}}^{\mathrm{DBDH}}(\lambda) := |\Pr[\mathcal{A}(P, A, B, C, e(P, P)^{abc}) = 1] - \Pr[\mathcal{A}(P, A, B, C, e(P, P)^{z}) = 1]|.$$

2.3 Indistinguishability Under Chosen-Plaintext Attack (IND-CPA)

For a public key encryption scheme $\Sigma = (\mathrm{Gen}, \mathrm{Enc}, \mathrm{Dec})$, indistinguishability under chosen plaintext attack (IND-CPA) is defined by the following game between an adversary $\mathcal{A}$ and a challenger C.

1. The challenger C generates key pair (pk, sk) by running the key generation algorithm $\mathrm{Gen}(1^k)$ on some security parameter k (e.g. a key size in bits). The challenger publishes pk to the adversary and retains sk.
2. The adversary $\mathcal{A}$ submits two equal-length chosen plaintexts $\{M_0, M_1\}$ to the challengers.
3. The challenger C selects a bit $b \in \{0, 1\}$ uniformly at random and sends the ciphertext to the adversary by running the encryption algorithm $\mathrm{Enc}(pk, m_b)$.

4. The adversary $\mathcal{A}$ outputs a guess for the value of b.

The advantage of the adversary $\mathcal{A}$ in this game is defined as $\mathrm{Adv}_{\Sigma,\mathcal{A}}^{\mathrm{IND-CPA}}(k) = \left|\Pr[b' = b] - \frac{1}{2}\right|$. The encryption scheme Σ is IND-CPA secure if $\mathrm{Adv}_{\Sigma,\mathcal{A}}^{\mathrm{IND-CPA}}(k)$ is negligible for any polynomial time adversary.

2.4 Ciphertext Policy Attribute-Based Encryption (CP-ABE)

In CP-ABE, a message is encrypted under an access structure W on attributes, and a secret key is associated with a set S of attributes. $S \vDash W$ represents that S satisfies W, and $S \nvDash W$ represents that S doesn't satisfy W. A ciphertext policy attribute-based encryption (CP-ABE) scheme consists of four fundamental algorithms: Setup, Extract, Encrypt, Decrypt.

Setup: The algorithm takes as input the security parameter λ and returns a public key PK and a master secret key MK.

Extract: The algorithm takes as input the master key MK and a set S of attributes. It returns a secret key SK associated with S.

Encrypt: The algorithm takes as input the public key PK, a message M and an access structure W. It returns a ciphertext CT with the property that a user with a secret key generated from attributes set S can decrypt CT if and only if $S \vDash W$.

Decrypt: The algorithm takes as input the ciphertext CT and the secret key SK. It returns the message M if $S \vDash W$, where S is the attribute set used to generate SK.

2.5 IND-CPA Security Game for CP-ABE

A CP-ABE scheme is said to be indistinguishability against chosen plaintext attacks (IND-CPA) in selective security model if no probabilistic polynomial-time adversaries have non-negligible advantage in the following game between an adversary $\mathcal{A}$ and a challenger C.

Init. The adversary $\mathcal{A}$ chooses the challenge access structure W^* and gives it to the challenger C.

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

Phase 1. The adversary $\mathcal{A}$ can the following query.

Ext query: The adversary $\mathcal{A}$ submits S to the challenger. Provided that $S \nvDash W^*$, the challenger answers with a secret key SK for S. This can be repeated adaptively.

Challenge. The adversary $\mathcal{A}$ submits two messages M_0 and M_1 of equal length. The challenger C chooses $\mu \in \{0, 1\}$ at random and encrypts M_μ under W^*. The resulting cipher CT^* text is given to the adversary $\mathcal{A}$.

Phase 2. The adversary $\mathcal{A}$ can continue to make queries as Phase 1.

Guess. Finally, the adversary $\mathcal{A}$ outputs a guess μ' of μ.

The advantage of an adversary $\mathcal{A}$ against the encryption scheme Σ is defined as $\mathrm{Adv}_{\Sigma,\mathcal{A}}^{\mathrm{IND-CPA}}(\lambda) = \left|\Pr[\mu' = \mu] - \frac{1}{2}\right|$. A ciphertext policy attribute-based encryption scheme is IND-CPA secure if all polynomial time adversaries have at most a negligible advantage in the IND-CPA game for CP-ABE.

3. System Model

3.1 Structure of the System

Figure 1 shows the architecture of secure data access control system in cloud storage with multi-authority. As shown in Fig. 1, the architecture consists of the following system entities:

1) User: A user downloads the shared data stored in the storage. If a user possesses a set of attributes satisfying the access policy of the encrypted data defined by the sender, and is not revoked in any of the attributes, then he will be able to decrypt the ciphertext and obtain the data, but he must not forward ciphertext to other users.
2) Data Owner: An owner encrypts the data and uploads it to cloud storage.
3) Key Authorities: They are key generation centers that generate public/secret parameters for CP-ABE. The key authorities consist of a central authority and multiple local authorities. A central authority manages a user and multiple local authorities manage attributes of a user.
4) Cloud Storage: This is an entity that stores data from owners. When users access the data, storage re-encrypts the shared data by re-encryption key and sends re-encrypted ciphertext to users. We assume the storage to be honest-but-curious. That is, it will honestly execute the tasks assigned by legitimate parties in the system. However, it would like to learn information of encrypted contents as much as possible.

3.2 Algorithm Definition

The system defined in Sect. 3.1 is composed of five algorithms. Key authorities run Auth.Setup and Auth.Ext. Sender runs DO.Enc. Storage runs C.ReEnc. User runs U.Dec.

Auth.Setup: It takes as input the security parameter λ and outputs the public key PK, the master secret key MK

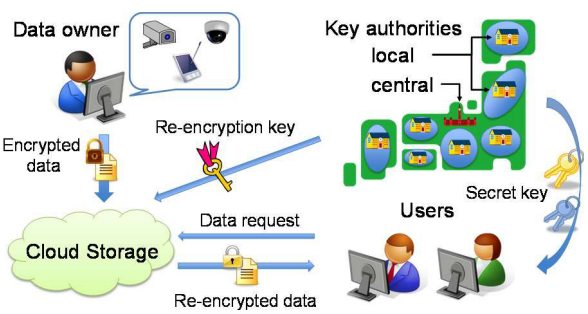

Fig. 1 Architecture of secure data access conrol system in cloud storage in multi-authority.

and the re-encryption key RK.

Auth.Ext: It takes as input the master secret key MK, a set S of attributes and the divided attribute universe $\mathcal{U}$, and outputs the secret key SK.

DO.Enc: It takes as input the public key PK, a message M and an access structure W, and outputs a ciphertext CT'.

C.ReEnc: It takes as input a ciphertext CT', a set S of attributes and the re-encrypted encryption key RK, and outputs a re-ciphertext CT.

U.Dec: It takes as input the secret key SK and a re-encrypted ciphertext CT. It outputs the message M if the attributes set S of SK satisfies the ciphertext access structure W.

3.3 Security Definitions

We prove that data confidentiality, collusion resistance, and forward secrecy are ensured. The attackers to break the confidentiality of the data are an unauthorized user and storage. Since we assume storage is honest, we do not consider active attacks from storage by colluding with unauthorized or revoked users. We define two attack models and security models as follows.

3.3.1 Security Definition in the Attack Model 1

In this model, we assume an attack by colluding unauthorized users and storage. IND-CPA security in this model is defined with the following game between an adversary $\mathcal{A}$ and a challenger C.

Init. The adversary $\mathcal{A}$ chooses the challenge access structure W^* and gives it to the challenger C.

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

Phase 1. The adversary $\mathcal{A}$ can the following query.

Ext query: The adversary $\mathcal{A}$ submits S to the challenger C. Provided that $S \not\models W^*$, the challenger C answers with a secret key SK for S. This can be repeated adaptively.

Challenge. The adversary $\mathcal{A}$ submits two messages M_0 and M_1 of equal length. The challenger chooses $\mu \in \{0, 1\}$ at random and generates CT'^* by encrypting M_μ under W^*. The challenger C runs C.ReEnc algorithm and submits CT^* to the adversary.

Phase 2. The adversary $\mathcal{A}$ can continue to make queries as Phase 1.

Guess. Finally, the adversary outputs a guess μ' of μ.

The advantage of an adversary $\mathcal{A}$ against the encryption scheme Σ is defined as $\mathrm{Adv}^{\mathrm{IND-CPA}}_{\Sigma,\mathcal{A}}(\lambda) = \left|\Pr[\mu' = \mu] - \frac{1}{2}\right|$. A ciphertext policy attribute-based encryption scheme is IND-CPA secure in the attack model 1 if all polynomial time adversaries have at most a negligible advantage in the IND-CPA game for CP-ABE.

3.3.2 Security Definition in the Attack Model 2

In this model, we assume an attack by the revoked user. IND-CPA security in this model is defined with the following game between an adversary $\mathcal{A}$ and a challenger C.

Init. The adversary $\mathcal{A}$ chooses the challenge access structure W^* and the revoked attribute x^*, and gives them to the challenger C. However, x^* satisfies $x^* \in I$ in W^*.

Setup. The challenger C runs the Setup algorithm and gives the adversary.

Phase 1. The adversary $\mathcal{A}$ can the following query.

Ext query: The adversary $\mathcal{A}$ submits S. The challenger C ansewers with a secret key SK for S. Provided that S satisfies $x^* \in S$ in the case $\underline{x^*} = x^*$ or $x^* \notin S$ in the case $\underline{x^*} = \neg x^*$. This can be repeated adaptively.

ReEnc query: The adversary $\mathcal{A}$ submits CT' and a set S^R of attributes. The challenger C answers the re-encrypted ciphertext CT. Provided that S^R satisfies $x^* \notin S^R$ in the case $\underline{x^*} = x^*$ or $x^* \in S^R$ in the case $\underline{x^*} = \neg x^*$. This can be repeated adaptively.

Challenge. The adversary $\mathcal{A}$ submits two messages M_0 and M_1 of equal length. The challenger C chooses $\mu \in \{0, 1\}$ at random and generates CT'^* by encrypting M_μ under W^*. The challenger C runs C.ReEnc algorithm and submits CT^* to the adversary.

Phase 2. The adversary $\mathcal{A}$ can continue to make queries as Phase 1.

Guess. Finally, the adversary $\mathcal{A}$ outputs a guess μ' of μ.

The advantage of an adversary $\mathcal{A}$ against the encryption scheme Σ is defined as $\mathrm{Adv}^{\mathrm{IND-CPA}}_{\Sigma,\mathcal{A}}(\lambda) = \left|\Pr[\mu' = \mu] - \frac{1}{2}\right|$. A ciphertext policy attribute-based encryption scheme is IND-CPA secure in the attack model 2 if all polynomial time adversaries have at most a negligible advantage in the IND-CPA game for CP-ABE.

4. Proposed Scheme

4.1 Overview

The proposed scheme is based on Cheung *et al.*'s scheme of CP-ABE [22]. In the proposed scheme, multiple authorities issue secret keys and can revoke the specified attribute. We divide attribute universe $\mathcal{U}$ and allocate to m local authorities.

In the proposed scheme, a ciphertext stored in storage cannot be decrypted with a secret key of a user as it is. When a user downloads a shared encrypted data from storage, storage re-encrypts the data and sends the re-encrypted data to the user. A user can obtain the original information by decrypting the re-encrypted data with his secret key. We assume that storage does not send a portion of ciphertext when a user downloads data in order that the user who is revoked the attribute cannot decrypt ciphertext. However, in this case, it is difficult to distinguish that ciphertext is incomplete in storage or correct ciphertext is destroyed in the way of sending. Therefore, in the proposed scheme, we avoid to be decrypted incomplete ciphertext.

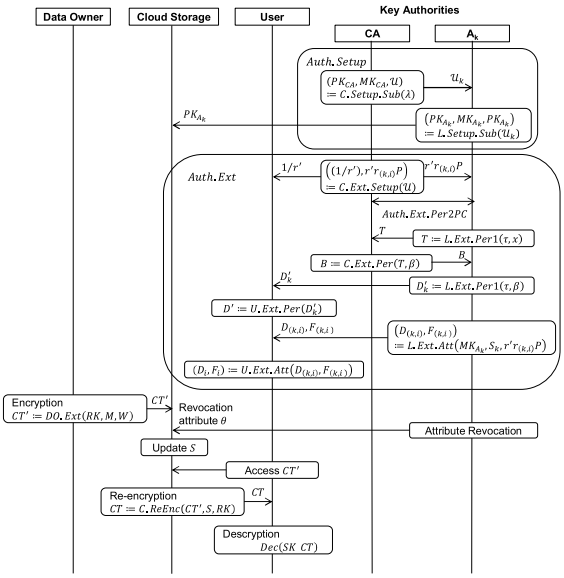

Fig. 2 Flow of proposed scheme

4.2 Definithions

In the proposed scheme, the number of attributes is n. Let $\mathcal{U} = \{1, \ldots, n\}$ be the attribute universe. Let CA be the central authority and $A = \{A_1, \ldots, A_m\}$ be the universe of local authorities. Let $\mathcal{U}_k = \{1, \ldots, n_k\}$ be the set of attributes managed by A_k. We assume each local authority manages a disjoint set of attributes such that $\mathcal{U}_k \cap \mathcal{U}_l = \emptyset$ for $k \neq l$. Let S_k be a attributes set of use's in $\mathcal{U}_k$. Each attribute would have three occurrences: *positive*, *negative*, and "*don't care*". We consider access structure W consisting of a single AND gate whose inputs are literals. This is denoted $\wedge_{i \in I} \underline{i}$, where $I \subseteq \mathcal{U}$ donates the set of attributes of interest and $\underline{i}$ is the literal of an attribute i, which can be positive (denoted by i) or negative (denoted by $\neg i$).

4.3 Algorithms

Figure 2 shows the flow of the proposed scheme. This flow includes procedure to revoke a particular attribute of a user before decryption. The attribute revocation procedure is shown in the next section.

The setup algorithm Auth.Setup consists of two algorithms, Algorithm 1-1. and Algorithm 1-2. CA runs Algorithm 1-1. Each A_k runs Algorithm 1-2.

Algorithm 1-1. C.Setup

INPUT: Security parameter λ (determing the size of the groups).
OUTPUT: CA's public key PK_{CA}, CA's master key MK_{CA}, attribute universe $\mathcal{U}$.

1. Generate a bilinear group $\mathbb{G}$ of prime order p with a generator P and a bilinear pairing e: $\mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$.
2. Randomly choose $\beta \in \mathbb{Z}_p$.
3. Compute $h = \beta P$.
4. Allocate attribute universe $\mathcal{U}$ into m local key authorities.
5. Return $PK_{CA} := (e, P, h)$, $MK_{CA} := \beta, \mathcal{U}_k$.

Algorithm 1-2. L.Setup

INPUT: Attribute universe $\mathcal{U}_k$.
OUTPUT: A_k's public key PK_{A_k}, A_k's master key MK_{A_k}, A_k's re-encryption key RK_{A_k}.

1. Randomly choose y_k, $t_{(k,1)}$, …, $t_{(k,3n_k)}$, $d_{(k,1)}$, …, $d_{(k,2n_k)} \in \mathbb{Z}_p$.
2. Compute as follows

$$Y_k := e(P, P)^{y_k}.$$

$$T_{(k,i)} := \begin{cases} d_{(k,i)}P & (1 \leq i \leq 2n_k). \\ t_{(k,i)}P & (2n_k + 1 \leq i \leq 3n_k). \end{cases}$$

$$rk_{(k,i)} := \frac{t_{(k,i)}}{d_{(k,i)}} \ (1 \leq i \leq 2n_k).$$

4. Return $PK_{A_k} := (Y_k, T_{(k,1)}, \ldots T_{(k,3n_k)})$, $MK_{A_k} := (y_k, d_{(k,1)}, \ldots d_{(k,2n_k)}, t_{(k,1)}, \ldots, t_{(k,3n_k)})$, $RK_{A_k} := (rk_{(k,1)}, \ldots rk_{(k,2n_k)})$

In the proposed scheme, a secret key component consists of a single personal key and multiple attribute keys. The proposed key generation algorithm is composed of the personal key generation followed by the attribute key generation algorithm. It exploits arithmetic secure 2PC protocol to eliminate the key escrow problem such that none of the authorities can determine the whole key components of users individually.

Algorithm 2-1. Ext.Per

INPUT: Attribute universe $\mathcal{U}$,
OUTPUT: Personal key component D'_k.
[CA]

1. For $i \in \mathcal{U}_k$,
   1.1 Randomly choose $r_{(k,i)} \in \mathbb{Z}_p$.
   1.2 Set $\gamma_k := \sum_{i=1}^{n_k} r_{(k,i)}$.
2. Set $r_t := \sum_{i=1}^{n_k} r_{(k,i)}$.
3. Randomly choose $r' \in \mathbb{Z}_p$.
4. Send $(1/r')$ to a user and $\{r'r_{(k,i)}P | i \in \mathcal{U}_k\}$ to A_k.

[A_k]

5. Compute $x = (y_k + \gamma_k)\beta$.
6. Randomly choose $\tau \in \mathbb{Z}_p$.
7. Compute $T = \left(\frac{x}{\tau}\right)P = \left\{\frac{(y_k+\gamma_k)\beta}{\tau}\right\}P$.

[CA]

8. Compute $B = \left(\frac{1}{\beta^2}\right)T = \left(\frac{y_k+\gamma_k}{\tau\beta}\right)P$.

[A_k]

9. Compute $D'_k = \tau B = \left(\frac{y_k+\gamma_k}{\beta}\right)P$.
10. Return personal key component D'_k.

A_k sends D'_k to a user securely. Then, the user computes its personal key component $D' = \prod_{k=1}^{m} D'_k = \left(\frac{y_1+\cdots+y_m+\gamma_1+\cdots+\gamma_m}{\beta}\right)P$.

Algprothm 2-2. Ext.Att

INPUT: Master secret key MK_{A_k}, a set of attributes S_k, $\{r'r_{(k,i)}P | i \in \mathcal{U}_k\}$.

OUTPUT: Attribute key component $D_{(k,i)}$, $F_{(k,i)}$.

1. For $i \in \mathcal{U}_k$, compute

$$D_{(k,i)} := \begin{cases} \frac{1}{t_{(k,i)}}(r' r_{(k,i)} P) & (i \in S_k). \\ \frac{1}{t_{(k,n_k+i)}}(r' r_{(k,i)} P) & (i \notin S_k). \end{cases}$$

$$F_{(k,i)} := \frac{1}{t_{(k,2n_k+i)}}(r' r_{(k,i)} P).$$

2. Return attribute key component $D_{(k,i)}$, $F_{(k,i)}$.

Each A_k gives attribute key component $\{D_{(k,i)}, F_{(k,i)} | i \in \mathcal{U}_k\}$ to the user. Then, the user computes $D_i = \left(\frac{1}{r'}\right) D_{(k,i)}$, $F_i = \left(\frac{1}{r'}\right) F_{(k,i)}$ for all its attribute key components and finally obtains its whole secret key $SK := (D', \{(D_i, F_i) | i \in \mathcal{U}\})$.

The encryption algorithm DO.Enc is described in Algorithm 3. It generates the ciphertext CT' to encrypt a plaintext M under the access structure W.

Algorithm 3. DO.Enc

INPUT: Public key PK, plaintext M, access structure W overt attribute universe $\mathcal{U}$.
OUTPUT: Ciphertext CT' under W.

1. Randomly choose a secret $s \in \mathbb{Z}_p$.
2. Compute $C := sh = s\beta P$, $\tilde{C} = M \cdot e(P, P)^{ys}$, where $\tilde{C} = M \cdot (Y_1 \times \cdots \times Y_m)^s = M \cdot e(P, P)^{ys}$.
3. For attribute universe $\mathcal{U}_k$ which A_k manages
   3.1 For $i \in \mathcal{U}_k$,
   3.2 Compute

$$C'_{(k,i)} := \begin{cases} sT_{(k,i)} & (i \in I \wedge \underline{i} = i). \\ sT_{(k,n_k+i)} & (i \in I \wedge \underline{i} = \neg i). \\ sT_{(k,2n_k+i)} & (i \notin I). \end{cases}$$

4. Return $CT' := (W, C, \tilde{C}, \{C_i | i \in \mathcal{U}\})$, where $C'_i = \cup_{k=1}^{m} C'_{(k,i)}$.

The re-encryption algorithm C.ReEnc is described in Algorithm 4. It generates the re-encrypted ciphertext CT to re-encrypt the ciphertext CT' according the attribute set S.

Algorithm 4. C.ReEnc

INPUT: Ciphertext $CT' = (W, C, \tilde{C}, \{C_i | i \in \mathcal{U}\})$, attribute set S, re-encryption key RK.
OUTPUT: Re-encrypted ciphertext CT.

1. Randomly choose a secret $s \in \mathbb{Z}_p$.
   1.1 In the case of $i \in S_k \wedge (i \in I \wedge \underline{i} = i)$,

$$\begin{aligned} C_{(k,,i)} &:= rk_{(k,i)} \cdot C'_{(k,i)} \\ &= \frac{t_{(k,i)}}{d_{(k,i)}} \cdot s \cdot d_{(k,i)} P = s \cdot t_{(k,i)} P. \end{aligned}$$

   1.2 In the case of $i \notin S_k \wedge (i \in I \wedge \underline{i} = \neg i)$,

$$\begin{aligned} C_{(k,i)} &:= rk_{(k,n_k+i)} \cdot C'_{(k,i)} \\ &= \frac{t_{(k,n_k+i)}}{d_{(k,n_k+i)}} \cdot s \cdot d_{(k,n_k+i)} P \\ &= s \cdot t_{(k,n_k+i)} P. \end{aligned}$$

   1.3 The other attributes,

$$C_{(k,i)} := C'_{(k,i)}.$$

2. Return re-encrypted ciphertext $CT := (W, C, \tilde{C}, \{C_i | i \in U\}$ where $C_i := \cup_{k=1}^{m} C_{(k,i)}$.

The decryption algorithm U.Dec is described in Algorithm 5. It generates the plaintext M to decrypt the ciphertext CT' according the secret key SK.

Algorithm 5. U.Dec

INPUT: Secret key SK, re-encrypted ciphertext CT.
OUTPUT: Plaintext M.

1. For attribute universe $\mathcal{U}_k$ which A_k manages,
   1.1 For $i \in \mathcal{U}_k$, compute as follows
   1.2 In the case of $i \in S_k \wedge \underline{i} = i$,

$$\begin{aligned} e(C_{(k,i)}, D_{(k,i)}) &:= e\left(s \cdot t_{(k,i)} P, \frac{r_{(k,i)}}{t_{(k,i)}} P\right) \\ &= e(P, P)^{s \cdot r_{(k,i)}}. \end{aligned}$$

   1.3 In the case of $i \notin S_k \wedge \underline{i} = \neg i$,

$$\begin{aligned} e(C_{(k,i)}, D_{(k,i)}) &:= e\left(s \cdot t_{(k,n_k+i)} P, \frac{r_{(k,i)}}{t_{(k,n_k+i)}} P\right) \\ &= e(P, P)^{s \cdot r_{(k,i)}}. \end{aligned}$$

   1.4 In the case of $i \notin I$,

$$\begin{aligned} e(C_{(k,i)}, F_{(k,i)}) &:= e\left(s \cdot t_{(k,2n_k+i)} P, \frac{r_{(k,i)}}{t_{(k,2n_k+i)}} P\right) \\ &= e(P, P)^{s \cdot r_{(k,i)}}. \end{aligned}$$

2. Compute

$$\begin{aligned} e(P, P)^{r_t s} &:= \prod_{k=1}^{m} e(P, P)^{\gamma_k s} \\ &= \prod_{k=1}^{m} \left(\prod_{i=1}^{n_k} e(P, P)^{s \cdot r_{(k,i)}} \right). \end{aligned}$$

$$\frac{\tilde{C}}{\frac{e(C, D')}{e(P, P)^{r_t s}}} = \frac{M e(P, P)^{ys}}{e(P, P)^{ys}} = M.$$

3. Return M.

4.4 Attribute Revocation Procedure

The attribute revocation is done by the following procedure:

1. A_k sends the user's revoked attribute i to storage.
2. Storage updates the attribute set of the user based on the revocation message.
3. When the user accesses the data in the storage, the storage runs re-encryption algorithm in accordance with an updated attribute set of the user, and sends the re-encrypted ciphertext to the user.

Because an updated set of attributes used for re-encryption does not satisfy access structure, the user cannot decrypt the

re-encrypted ciphertext.

5. Security Proof

5.1 Security Proof in the Attack Model 1

Theorem 1. *If a probabilistic polynomial-time adversary can win the CP-ABE game with non-negligible advantage in the attack model 1, then we can construct a simulator that can distinguish a DBDH tuple from a random tuple with non-negligible advantage.*

Proof. Suppose adversary $\mathcal{A}$ can win the CP-ABE game in the attack model 1 with non-negligible advantage ϵ. We construct a simulator $\mathcal{B}$ that can distinguish a DBDH tuple from a random tuple with non-negligible advantage $\epsilon/2$. Let e: $\mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ be an efficiently computable bilinear map, where G has prime order p. First the DBDH challenger C selects at random: $a,\ b,\ c,\ z \in \mathbb{Z}_p$, $v \in \{0,1\}$ and generator $P \in \mathbb{G}$. It defines Z to be $e(P,P)^{abc}$ if $v = 0$ and $e(P,P)^z$ otherwise. The challenger C then gives the simulator $\mathcal{B}$ $\langle P, A, B, C, Z\rangle = \langle P, aP, bP, cP, Z\rangle$. The simulator $\mathcal{B}$ plays the role of challenger in the CP-ABE game.

-*Init.* The adversary $\mathcal{A}$ submits the challenge access structure $W^* = \wedge_{i\in I}\underline{i}$ to the simulator $\mathcal{B}$.

-*Setup.* The simulator $\mathcal{B}$ generate a public key PK. $\mathcal{B}$ sets Y to be $e(A,B) = e(P,P)^{ab}$. For each $i \in \mathcal{U}$, $\mathcal{B}$ chooses random $\alpha_i,\ \beta_i,\ \gamma_i \in \mathbb{Z}_p$. It now constructs T_i, T_{n+i} and T_{2n+i} as in Table 1. Then, $\mathcal{B}$ chooses $\kappa \in \mathbb{Z}_p$ at random and sets h to be κP. Finally, $\mathcal{B}$ chooses a re-encryption key $rk_i \in \mathbb{Z}_p$ $(1 \le i \le 2n)$ at random. $\mathcal{B}$ submits the public key PK and the re-encryption key RK to the adversary $\mathcal{A}$.

-*Phase 1.* The adversary $\mathcal{A}$ can the following query.

-*Ext query.* $\mathcal{A}$ submits a set $S \subseteq \mathcal{U}$, where $S \not\models W^*$. There must exist $j \in I$ such that: either $j \in S$ and $\underline{j} = \neg j$, or $j \notin S$ and $\underline{j} = j$. $\mathcal{B}$ chooses such j. Without loss of generality, assume that $j \notin S$ and $\underline{j} = j$. For every $i \in \mathcal{U}$, $\mathcal{B}$ chooses $r'_i \in \mathbb{Z}_p$ at random. It then sets $r_j := -ab + r'_j \cdot b$ and, for every $i \neq j$, it sets $r_i := r'_i \cdot b$. Finally, it sets $r := \sum_{i=1}^n r_i = -ab + \sum_{i=1}^n r'_i \cdot b$. The D' component of the secret key can be computed as:

$$D' := \left(\frac{1}{\kappa}\right) \cdot \prod_{i=1}^n r'_i \cdot B = \frac{1}{\kappa} \cdot \left(\sum_{i=1}^n r'_i \cdot b\right) P = \left(\frac{ab + r}{\kappa}\right) P.$$

Recall that $j \in I \setminus S$ and $\underline{j} = j$, therefore the D_j component can be computed as:

$$D_j := \left(-\frac{1}{\beta_j}\right) A \cdot \frac{1}{\beta_j} P = \frac{-ab + r'_j \cdot b}{b \cdot \beta_j} P = \frac{r_j}{b \cdot \beta_j} P.$$

For $i \neq j$, we have a few cases:
In the case of $i \in S$.
(1) $i \in I \wedge \underline{i} = i$.

Table 1 Computation of Public key Components of T_i.

	$i \in I$		$i \notin I$
	$\underline{i} = i$	$\underline{i} = \neg i$	
T_i	$\alpha_i P$	$\alpha_i B$	$\alpha_i B$
T_{n+i}	$\beta_i B$	$\beta_i P$	$\beta_i B$
T_{2n+i}	$\gamma_i B$	$\gamma_i B$	$\gamma_i P$

$$D_i := \frac{r'_i}{\alpha_i \cdot rk_i} B = \frac{r_i}{\alpha_i \cdot rk_i} P.$$

(2) $(i \in I \wedge \underline{i} = \neg i) \vee i \notin$

$$D_i := \frac{r'_i}{\alpha_i \cdot rk_i} B = \frac{r_i}{b \cdot \alpha_i \cdot rk_i} P.$$

In the case of $i \notin S$.
(1) $(i \in I \wedge \underline{i} = i) \vee i \notin I$.

$$D_i := \frac{r'_i}{\beta_i \cdot rk_{n+i}} B = \frac{r_i}{b \cdot \beta_i \cdot rk_{n+i}} P.$$

(2) $i \in I \wedge \underline{i} = \neg i$.

$$D_i := \frac{r'_i}{\beta_i \cdot rk_{n+i}} B = \frac{r_i}{\beta_i \cdot rk_{n+i}} P.$$

The F_i components are computed as follows. First,

$$F_j := \left(-\frac{1}{\gamma_j}\right) A \cdot \frac{r'_j}{\gamma_j} P = \frac{-ab + r'_j \cdot b}{\gamma_j \cdot b} P = \frac{r_j}{b \cdot \gamma_j} P.$$

For $i \neq j$, we have two cases.
(1) $i \in I$. $F_i := \frac{r'_i}{\gamma_i} P = \frac{r_i}{b \cdot \gamma_i} P$.
(2) $i \notin I$. $F_i := \frac{r'_i}{\gamma} B = \frac{r_i}{\gamma_i} P$.
From the above, $\mathcal{B}$ submits the secret key to $\mathcal{A}$.

-*Challenge.* $\mathcal{A}$ submits two equal length messages M_0 and M_1. $\mathcal{B}$ chooses $\mu \in \{0,1\}$ at random and sets $\tilde{C} := M_\mu \cdot Z$. $\mathcal{B}$ gives $\mathcal{A}$ the following re-encrypted ciphertext CT^*.

$$CT^* := \left(\begin{array}{c} W, \tilde{C}, C, \{rk_i \alpha_i C | i \in I \wedge \underline{i} = i\}, \\ \{rk_{n+i} \beta_i C | i \in I \wedge \underline{i} = \neg i\},\ \{\gamma_i C | i \notin I\} \end{array} \right).$$

-*Phase 2.* Same as Phase 1.

-*Guess.* $\mathcal{A}$ submits a guess μ' of μ. If $\mu' = \mu$, $\mathcal{B}$ will output $v' = 0$, meaning that (A, B, C, Z) is a valid DBDH-tuple; otherwise, $\mathcal{B}$ outputs $v' = 1$, indicating that (A, B, C, Z) is just a random 4-tuple.

In the case of $v = 1$, the adversary obtains no information about μ. Therefore $\mu' \neq \mu$ holds with probability exactly $\frac{1}{2}$, regardless of the distribution on μ'. We thus have $\Pr[\mu' \neq \mu | v = 1] = \frac{1}{2}$. $\mathcal{B}$ just randomly guesses $v' = 1$ when $\mu \neq \mu'$, we have $\Pr[v' = v | v = 1] = \frac{1}{2}$. In the case of $v = 0$, then CT is a valid ciphertext, in which case the advantage of $\mathcal{A}$ is $\mathrm{Adv}^{\mathrm{IND-CPA}}_{\Sigma,\mathcal{A}}(\lambda)$. We thus have $\Pr[\mu' = \mu | v = 0] = \frac{1}{2} + \mathrm{Adv}^{\mathrm{IND-CPA}}_{\Sigma,\mathcal{A}}(\lambda)$. Since $\mathcal{B}$ guesses $v' = 0$ whether $\mu = \mu'$, we have $\Pr[v' = v | v = 0] = \frac{1}{2} + \mathrm{Adv}^{\mathrm{IND-CPA}}_{\Sigma,\mathcal{A}}(\lambda)$. The overall advantage of $\mathcal{B}$ in the DBDH game is

$$\mathrm{Adv}^{\mathrm{DBDH}}_{\mathcal{B}}(\lambda) = \left| \Pr[v' = v] - \frac{1}{2} \right|$$

$$= \left| \frac{1}{2}\Pr[v' = v|v = 0] + \frac{1}{2}\Pr[v' = v|v = 1] - \frac{1}{2} \right|$$
$$= \left| \frac{1}{2}\left(\frac{1}{2} + \mathrm{Adv}^{\mathrm{IND-CPA}}_{\Sigma,\mathcal{A}}(\lambda) \right) + \frac{1}{2} \cdot \frac{1}{2} - \frac{1}{2} \right|$$
$$= \frac{1}{2}\mathrm{Adv}^{\mathrm{IND-CPA}}_{\Sigma,\mathcal{A}}(\lambda).$$

■

5.2 Security Proof in the Attack Model 2

Theorem 2. *If a probabilistic polynomial-time adversary can win the CP-ABE game with non-negligible advantage in the attack model 2, then we can construct a simulator that can distinguish a DBDH tuple from a random tuple with non-negligible advantage.*

Proof. Suppose adversary $\mathcal{A}$ can win the CP-ABE game in the attack model 1 with non-negligible advantage ϵ. We construct a simulator $\mathcal{B}$ that can distinguish a DBDH tuple from a random tuple with non-negligible advantage $\epsilon/2$. Let e: $\mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ be an efficiently computable bilinear map, where G has prime order p. First the DBDH challenger C selects at random: $a, b, c, d \in \mathbb{Z}_p$, $v \in \{0, 1\}$ and generator $P \in \mathbb{G}$. It defines Z to be $e(P, P)^{abc}$ if $v = 0$ and $e(P, P)^z$ otherwise. The challenger C then gives the simulator $\mathcal{B}$ $\langle P, A, B, C, Z\rangle = \langle P, aP, bP, cP, Z\rangle$. The simulator $\mathcal{B}$ plays the role of challenger in the CP-ABE game.

-Init. The adversary $\mathcal{A}$ submits the challenge access structure $W^* = \wedge_{i\in I}\underline{i}$ and the revoked attribute $x^* \in \mathcal{U}$, where x^* satisfies $x^* \in I$, to the simulator $\mathcal{B}$.

-Setup. The simulator $\mathcal{B}$ generate a public key. $\mathcal{B}$ sets Y to be $e(A, B) = e(P, P)^{ab}$. For each $i \in \mathcal{U}$, $\mathcal{B}$ chooses random $\alpha_i, \beta_i, \gamma_i \in \mathbb{Z}_p$. It now constructs T_i, T_{n+i} and T_{2n+i} as in Table 1. Then, $\mathcal{B}$ chooses $\kappa \in \mathbb{Z}_p$ at random and sets h to be κP. Additionally, $\mathcal{B}$ chooses $z_{x^*} \in \mathbb{Z}_p$ at random and sets $rk_{x^*} := \frac{(z_{x^*}\cdot b)}{\alpha_{x^*}}$ if $x^* \in I \wedge \underline{x^*} = x^*$ or $rk_{x^*} := \frac{(z_{x^*}\cdot b)}{\beta_{x^*}}$ if $x^* \in I \wedge \underline{x^*} = \neg x^*$. Finally, for every $i \neq x^*$, $\mathcal{B}$ chooses a re-encryption key $rk_i \in \mathbb{Z}_p$ $(1 \leq i \leq 2n)$ at random. $\mathcal{B}$ submits a public key PK and a re-encryption key RK to the adversary $\mathcal{A}$.

-Phase 1. The adversary $\mathcal{A}$ can the following query.

-Ext Query. $\mathcal{A}$ submits a sets $S \subseteq \mathcal{U}$. Provided that S satisfies $x^* \in S$ in the case $\underline{x^*} = x^*$ or $x^* \notin S$ in the case $\underline{x^*} = \neg x^*$. $\mathcal{B}$ chooses such j. Without loss of generality, assume that $j \notin S$ and $\underline{j} = j$. For every $i \in \mathcal{U}$, $\mathcal{B}$ chooses $r'_i \in \mathbb{Z}_p$ at random. It then sets $r_j := -ab + r'_j \cdot b$ and, for every $i \neq j$, it sets $r_i := r'_i \cdot b$. Finally, it sets $r := \sum_{i=1}^n r_i = -ab + \sum_{i=1}^n r'_i \cdot b$. The D' component of the secret key can be computed as:

$$D' := \left(\frac{1}{\kappa}\right) \cdot \prod_{i=1}^n r'_i \cdot B$$
$$= \frac{1}{\kappa} \cdot \left(\sum_{i=1}^n r'_i \cdot b \right) P = \left(\frac{ab + r}{\kappa} \right) P.$$

For $i = x^*$, the D_{x^*} component can be computed as:
In the case of $x^* \in I \wedge \underline{x^*} = x^*$

$$D_{x^*} := \left(-\frac{1}{z_{x^*}}\right) A \cdot \left(\frac{r'_{x^*}}{z_{x^*}}\right) P$$
$$= \left(\frac{-ab + r'_{x^*} \cdot b}{z_{x^*} \cdot b} \right) P = \left(\frac{r'_{x^*}}{\alpha_{x^*} \cdot rk_{x^*}} \right) P.$$

In the case of $x^* \in I \wedge \underline{x^*} = \neg x^*$

$$D_{x^*} := \left(-\frac{1}{z_{x^*}}\right) A \cdot \left(\frac{r'_{x^*}}{z_{x^*}}\right) P$$
$$= \left(\frac{-ab + r'_{x^*} \cdot b}{z_{x^*} \cdot b} \right) P = \left(\frac{r'_{x^*}}{\beta_{x^*} \cdot rk_{x^*}} \right) P.$$

For $i \neq x^*$, we habe a few cases.
In the case of $i \in S$.
(1) $i \in I \wedge \underline{i} = i$.

$$D_i := \frac{r'_i}{\alpha_i \cdot rk_i} B = \frac{r_i}{\alpha_i \cdot rk_i} P.$$

(2) $(i \in I \wedge \underline{i} = \neg i) \vee i \notin I$.

$$D_i := \frac{r'_i}{\alpha_i \cdot rk_i} B = \frac{r_i}{b \cdot \alpha_i \cdot rk_i} P.$$

In the case of $i \notin S$.
(1) $(i \in I \wedge \underline{i} = i) \vee i \notin I$.

$$D_i := \frac{r'_i}{\beta_i \cdot rk_{n+i}} B = \frac{r_i}{b \cdot \beta_i \cdot rk_{n+i}} P.$$

(2) $i \in I \wedge \underline{i} = \neg i$.

$$D_i := \frac{r'_i}{\beta_i \cdot rk_{n+i}} B = \frac{r_i}{\beta_i \cdot rk_{n+i}} P.$$

The F_i components are computed as follows. First,

$$F_{x^*} := \left(-\frac{1}{\gamma_{x^*}}\right) A \cdot \frac{r'_{x^*}}{\gamma_{x^*}} P = \frac{-ab + r'_{x^*} \cdot b}{\gamma_{x^*} \cdot b} P$$
$$= \frac{r_{x^*}}{b \cdot \gamma_{x^*}} P.$$

For $i \neq x^*$, we have two cases.
(1) $i \in I$. $F_i := \frac{r'_i}{\gamma_i} P = \frac{r_i}{b\cdot\gamma_i} P$.
(2) $i \notin I$. $F_i := \frac{r'_i}{\gamma} B = \frac{r_i}{\gamma_i} P$.
From the above, $\mathcal{B}$ submits the secret key to A.

-ReEnc Query. $\mathcal{A}$ submits a ciphertext CT' and a set S^R, where S^R satisfies $x^* \notin S^R$ in the case $\underline{x^*} = x^*$ or $x^* \in S^R$ in the case $\underline{x^*} = \neg x^*$. For every $i \in S^R \wedge (i \in I \wedge \underline{i} = i)$ or $i \notin S^R \wedge (i \in I \wedge \underline{i} = \neg i)$, $\mathcal{B}$ computes $C_i := rk_i \cdot C'_i$ and submits the re-encrypted ciphertext CT to $\mathcal{A}$.

-Challenge. $\mathcal{A}$ submits two equal length messages M_0 and M_1. $\mathcal{B}$ chooses $\mu \in \{0, 1\}$ at random and sets $\tilde{C} := M_\mu \cdot Z$. $\mathcal{B}$ gives $\mathcal{A}$ the following re-encrypted ciphertext CT^*.

In the case of $x^* \in I \wedge \underline{x^*} = x^*$.

$$CT^* := \left(\begin{array}{c} W, \tilde{C}, C, \{rk_i \alpha_i C | i \neq x^* \wedge (i \in I \wedge \underline{i} = i)\}, \\ \{\beta_i C | i = x^* \wedge i \in I \wedge \underline{i} = \neg i\}, \\ \{rk_{n+i} \beta_i C | i \in I \wedge \underline{i} = \neg i\}, \{\gamma_i C | i \notin I\} \end{array} \right).$$

In the case of $x^* \in I \wedge \underline{x^*} = \neg x^*$.

$$CT^* := \left(\begin{array}{c} W, \tilde{C}, C, \{rk_i \alpha_i C | i \neq x^* \wedge (i \in I \wedge \underline{i} = i)\}, \\ \{\alpha_i C | i = x^* \wedge i \in I \wedge \underline{i} = \neg i\}, \\ \{rk_{n+i} \beta_i C | i \in I \wedge \underline{i} = \neg i\}, \{\gamma_i C | i \notin I\} \end{array} \right).$$

-Phase 2. Same as Phase 1.

-Guess. $\mathcal{A}$ submits a guess μ' of μ. If $\mu' = \mu$, $\mathcal{B}$ will output $\nu' = 0$, meaning that (A, B, C, Z) is a valid DBDH-tuple; otherwise, $\mathcal{B}$ outputs $\nu' = 1$, indicating that (A, B, C, Z) is just a random 4-tuple.

In the case of $\nu = 1$, the adversary obtains no information about μ. Therefore $\mu' \neq \mu$ holds with probability exactly $\frac{1}{2}$, regardless of the distribution on μ'. We thus have $\Pr[\mu' \neq \mu | \nu = 1] = \frac{1}{2}$. $\mathcal{B}$ just randomly guesses $\nu' = 1$ when $\mu \neq \mu'$, we have $\Pr[\nu' = \nu | \nu = 1] = \frac{1}{2}$. In the case of $\nu = 0$, then CT is a valid ciphertext, in which case the advantage of $\mathcal{A}$ is $\mathrm{Adv}^{\mathrm{IND-CPA}}_{\Sigma,\mathcal{A}}(\lambda)$. We thus have $\Pr[\mu' = \mu | \nu = 0] = \frac{1}{2} + \mathrm{Adv}^{\mathrm{IND-CPA}}_{\Sigma,\mathcal{A}}(\lambda)$. Since $\mathcal{B}$ guesses $\nu' = 0$ whether $\mu = \mu'$, we have $\Pr[\nu' = \nu | \nu = 0] = \frac{1}{2} + \mathrm{Adv}^{\mathrm{IND-CPA}}_{\Sigma,\mathcal{A}}(\lambda)$. The overall advantage of $\mathcal{B}$ in the DBDH game is

$$\begin{aligned} &\mathrm{Adv}^{\mathrm{DBDH}}_{\mathcal{B}}(\lambda) \\ &= \left| \Pr[\nu' = \nu] - \frac{1}{2} \right| \\ &= \left| \frac{1}{2} \Pr[\nu' = \nu | \nu = 0] + \frac{1}{2} \Pr[\nu' = \nu | \nu = 1] - \frac{1}{2} \right| \\ &= \left| \frac{1}{2} \left(\frac{1}{2} + \mathrm{Adv}^{\mathrm{IND-CPA}}_{\Sigma,\mathcal{A}}(\lambda) \right) + \frac{1}{2} \cdot \frac{1}{2} - \frac{1}{2} \right| \\ &= \frac{1}{2} \mathrm{Adv}^{\mathrm{IND-CPA}}_{\Sigma,\mathcal{A}}(\lambda). \end{aligned}$$

■

6. Discussion

6.1 Requirement

In this section, we verify whether the proposed scheme meets five requirements (1: the authority architecture, 2: revocation granularity, 3: security, 4: key update, 5: the length of ciphertext) described in Sect. 1.

1) Requirement 1: The value $D_{(k,i)}$ and $F_{(k,i)}$ are set for each attribute $i \in \mathcal{U}_k$ and each A_k manages. Especially, $D_{(k,i)}$ is set based on the attribute set S_k, A_k can define S_k. A_k sends S_k to the storage, it can be regarded as a whole attribute set S by collecting S_k from all A_k. Furthermore, after key generation, A_k can revoke the specified attribute without key update. Thus, the proposed scheme meets the requirement 1.

2) Requirement 2: When we want to revoke the specified attribute, A_k sends information of attribute to the storage and the storage updates the attribute set S immediately. The ciphertext stored in the storage cannot be decrypted. Re-encryption based on the attribute set S is required. The re-encryption is performed when user accesses the new data. There is no possibility that unauthorized user can decrypt because the ciphertext is re-encrypted by the latest set of attributes. Thus, the proposed scheme meets the requirements 2.

3) Requirement 3: We define attack models and security models in Sect. 3.3. We prove the proposed scheme is IND-CPA secure under the DBDH assumption.

4) Requirement 4: Attribute revocation of user is carried out in accordance with Sect. 4.4. During attribute revocation, only S is updated, secret key is not updated. Thus, the proposed scheme meets the requirements 4.

5) Requirement 5: In the proposed scheme, all attributes are associated with access structure, so the size of the ciphertext are fixed length in the same system regardless of the number of attributes required to decrypt. Thus, the proposed scheme meets the requirements 5.

6.2 Comparison of Computation Amount

Table 2 shows comparison of computation amount among CP-ABE schemes. In JLWW13 and YJ14, the computation amount of encryption is proportional to the number of attributes appeared in the access structure, which is represented as |I|. Especially, in YJ14, the computation amount of encryption becomes larger than that in the proposed scheme when |I| is larger than 1/4 of the number of attributes in system. |I| would be large value because an access structure becomes complex and large scale in order to perform fine-grained access control. On the other hand, in CN07, YWRL10, and the proposed scheme, the computation amount of encryption is the same and fixed value even if |I| is large value. In addition, the table shows that CN07, YWRL10, and YJ14 need computation of key update and the proposed scheme does not need computation of key update. Users cannot update their secret key when the key is embedded in IoT devices. Therefore, those advantages would be significant.

The computation amount of key generation is the largest among CP-ABE schemes. However, generating keys is only once. For example, in IoT devices, keys can be embedded at the same time when they are manufactured. Therefore, the disadvantage would be insignificant.

6.3 Comparison among the CP-ABE Scheme

Table 3 shows whether meets the five requirements, expressiveness of access structure and whether it is possible to specify a negative attribute in access structure.

First, single authority issues secret key of users in

Table 2 Comparison of size and computation amount

		CN07 [22]	YWRL10 [37]	JLWW13 [16]	YJ14 [15]	Proposed	
Size	Public key	$(3\|U\|+1)\|G\|+\|G_T\|$	$(3\|U\|+1)\|G\|+\|G_T\|$	$\|G\|+\|G_T\|$	$(2+2\|U\|+m)\|G\|$ $+m\times\|G_T\|$	$(3\|U\|+2+m)\|G\|+2m\|G_T\|$	
	Secret key	$(2\|U\|+1)\|G\|$	$(2\|U\|+1)\|G\|$	$(2\|S\|+1)\|G\|$	$(m+m_u+\|S\|)\|G\|$	$(2\|U\|+1)\|G\|$	
	Ciphertext	$(\|U\|+1)\|G\|+\|G_T\|$	$(\|U\|+1)\|G\|+\|G_T\|$	$2\|I\|\|G\|+2\|G_T\|$	$(4+4\|I\|)\|G\|+\|G_T\|$	$(\|U\|+1)\|G\|$	
	Re-encryption key	$2r\|U\|\|Z_p\|$	$2r\|U\|\|Z_p\|$	—	$2\|R_{CT}\|\times\|Z_p\|$	$2\|U\|\|Z_p\|$	
Computation amount	Encryption	$(\|U\|+2)C$	$(\|U\|+2)C$	$(2+2\|I\|)C$	$(3+4\|I\|)C$	$(\|U\|+2)C$	
	Key generation	$(2\|U\|+1)C$	$(2\|U\|+1)C$	$(2+2m)C$	$\|S\|C$	A_k	$(2\|U\|+2m)C$
						CA	$(\|U\|+m)C$
						Users	$2\|U\|C$
	Re-encryption	$\|R_{CT}\|C$	$\|R_{CT}\|C$	$(2\|R\|+1)Ce$ $+(2\|R\|+3+2\|I\|)C$	$2\|R_{CT}\|C$	$\|R\|C$	
	Key update	$\|R_{SK}\|C$	$\|R_{SK}\|C$	—	$\|R_{SK}\|C$	—	
	Decryption	$(\|U\|+1)C_e+(\|U\|+1)C$	$(\|U\|+1)C_e+(\|U\|+1)C$	$(2\|R\|+1)Ce$ $+(2\|R\|+1)C$	$(2m_{CT}+4\|R\|)Ce$ $+(m_{CT}+\|R\|)C$	$(\|U\|+1)C_e+(\|U\|+1)C$	

C: the exponentiation in group G, C_e: bilinear pairing,$|G|$: bit size of an element in G, $|G_T|$: bit size of element in G_T
$|U|$: the number of attributes in the system, $|S|$: the number of attributes associated with secret key of a user, m: the number of the local authorities
m_u: the number of the local authorities which manage user's attributes, r: the number of the attribute revocation events
$|R|$: the number of attributes required for decryption, $|R_{CT}|$:the number of the update attributes in ciphertext, $|R_{SK}|$: the number of the update attributes in secret key,
$|I|$: the number of attributes appeared in the access structure, A_k: a local authority k, CA: central authority

Table 3 Comprehensive comparison of requirements which the CP-ABE schemes meet

	CN07 [22]	YWRL10 [37]	JLWW13 [16]	YJ14 [15]	Proposed
Authority	single	single	multiple	multiple	multiple
Revocation	—	immediate attribute-level user revocation	immediate system-level user revocation	immediate attribute-level user revocation	immediate attribute-level user revocation
Security in model 1	IND-CPA(DBDH)	IND-CPA(DBDH)	IND-CPA(DBDH)	IND-CPA(q-DBHE)	IND-CPA(DBDH)
Security in model 2	—	—	—	informal	IND-CPA(DBDH)
Key Update	—	needed (proxy server perform)	—	needed (user perform)	not needed
Length of Ciphertext	fixed	fixed	not fixed	not fixed	fixed
Expressiveness	AND	AND	AND,OR,k-out-of-n	AND,OR	AND
Negative attribute in access structure	configurable	configurable	not configurable	not configurable	configurable

IND-CPA: indistinguishability against chosen plaintext attacks, DBDH: Decisional Bilinear Diffie-Hellman Assumption
q-DBHE: decisional q-parallel Bilinear Diffie-Hellman Exponent assumption

CN07 and YWRL10 while in JLWW13, YJ14 and the proposed scheme, multiple authorities issue a secret key. Then, the immediate revocation of a user can be done except CN07. In JLWW13, user revocation is executed by re-encrypting by access structure including subtree which is associated revoked users. YWRL10, YJ14, and the proposed scheme can assign a specific attribute to revoke a user. As for the security, only the proposed scheme have been shown that it is IND-CPA secure against an attack by unauthorized users, the cloud storage, and a revoked user. The proxy server performs key update in YWRL10. A user update secret key when he receives the update key from key authorities in YJ14. In JLWW13 and YJ14, length of ciphertext is not fixed, but in CN07, YWRL10, and the proposed scheme, it is fixed. CN07, YWRL10 and the proposed scheme only allow AND gate. In JLWW13, access structure is described with an access tree, so it allows AND, OR, and k-out-of-n gate. In YJ14, access structure is described with a LSSS structure, so it allows AND and OR gate. Therefore, the only proposed scheme meets all requirements.

7. Conclusion

In this paper, a ciphertext-policy attribute-based encryption scheme with attribute revocation and forward secrecy is applied to cloud storage. Multiple independent key authorities were proposed to manage the attributes of users. When the attributes of users are specified to be revoked, the revocation becomes effective immediately. After attributions are revoked, there is no necessary to update the secret key in the terminal of users because the storage node updates attribute sets and re-encrypts ciphertext according to new attribute sets. The length of key size and ciphertext size is able to be fixed. In addition, the proposed scheme is IND-CPA secure in DBDH assumption under the standard model.

In the proposed scheme, the access structure is expressed with only one AND gate. The access structure with more free degree will be investigated in our future work.

Acknowledgments

This work was supported by JSPS KAKENHI Grant Number 16K00184 and 22700067.

References

[1] V. Turner, J.F. Gantz, D. Reinsel, and S. Minton, "The digital universe of opportunities: rich data and the increasing value of the internet of things," IDC Analyze the Future, 2014.
[2] Y. Ma, J. Rao, W. Hu, X. Meng, X. Han, Y. Zhang, Y. Chai, and C. Liu, "An efficient index for massive IOT data in cloud environment," Proc. 21st ACM international conference on Information and

knowledge management, pp.2129–2133, 2012.
[3] C. Liu, C. Yang, X. Zhang, and J. Chen, "External integrity verification for outsourced big data in cloud and IoT: A big picture," Future Generation Computer Systems, vol.49, pp.58–67, 2015.
[4] C. Liu, X. Zhang, C. Yang, and J. Chen, "CCBKE—Session key negotiation for fast and secure scheduling of scientific applications in cloud computing," Future Generation Computer Systems, vol.29, no.5, pp.1300–1308, 2013.
[5] K. Zhang, X. Zhou, Y. Chen, X. Wang, and Y. Ruan, "Sedic: privacy-aware data intensive computing on hybrid clouds," Proc. 18th ACM conference on Computer and communications security, pp.515–526, 2011.
[6] C. Yang, X. Zhang, C. Zhong, C. Liu, J. Pei, K. Ramamohanarao, and J. Chen, "A spatiotemporal compression based approach for efficient big data processing on cloud," Journal of Computer and System Sciences, vol.80, no.8, pp.1563–1583, 2014.
[7] Y. Yang, J.K. Liu, K. Liang, K.-K.R. Choo, and J. Zhou, "Extended proxy-assisted approach: achieving revocable fine-grained encryption of cloud data," European Symposium on Research in Computer Security, pp.146–166, 2015.
[8] G. Wang, Q. Liu, J. Wu, and M. Guo, "Hierarchical attribute-based encryption and scalable user revocation for sharing data in cloud servers," computers & security, vol.30, no.5, pp.320–331, 2011.
[9] G. Wang, Q. Liu, and J. Wu, "Hierarchical Attribute-Based Encryption for Fine-Grained Access Control in Cloud Storage Services," Proc. 17th ACM conference on Computer and communications security, pp.735–737, 2010.
[10] S. Yu, C. Wang, K. Ren, and W. Lou, "Achieving Secure, Scalable, and Fine-grained Data Access Control in Cloud Computing," Proc. INFOCOM'10, pp.1–9, 2010.
[11] S. Ruj, A. Nayak, and I. Stojmenovic, "DACC: Distributed Access Control in Clouds," Proc. 2011 IEEE 10th International Conference on Trust, Security and Privacy in Computing and Communications, pp.91–98, 2011.
[12] K. Yang and X. Jia, "Attributed-based access control for multi-authority systems in cloud storage," Proc. ICDCS 2012, pp.536–545, 2012.
[13] K. Yang, X. Jia, and K. Ren, "Attribute-based fine-grained access control with efficient revocation in cloud storage systems," AsiaCCS'13, pp.523–528, 2013.
[14] K. Yang, X. Jia, K. Ren, B. Zhang, and R. Xie, "DAC-MACS: Effective Data Access Control for Multiauthority Cloud Storage Systems," IEEE Trans. Inf. Forensics Security, vol.8, no.11, pp.1790–1801, 2013.
[15] K. Yang and X. Jia, "Expressive, efficient, and revocable data access control for multi-authority cloud storage," IEEE Trans. Parallel Distrib. Syst., vol.25, no.7, pp.1735–1744, 2014.
[16] T. Jung, X.-Y. Li, Z. Wan, and M. Wan, "Privacy preserving cloud data access with multi-authorities," INFOCOM'13, pp.2625–2633, 2013.
[17] A. Sahai and B. Waters, "Fuzzy identity-based encryption," Proc. Eurocrypt, pp.457–473, 2005.
[18] A. Shamir, "Identity-Based Cryptosystems and Signature Schemes," Proc. CRYPTO 84, LNCS, vol.196, pp.47–53, 1984.
[19] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," Proc. ACM Conf. Comput. Commun. Security, pp.89–98, 2006.
[20] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," Proc. IEEE Symp. Security Privacy, pp.321–334, 2007.
[21] R. Ostrovsky, A. Sahai, and B. Waters, "Attribute-based encryption with non-monotonic access structures," Proc. ACM Conf. Comput. Commun. Security, pp.195–203, 2007.
[22] L. Cheung and C. Newport, "Provably secure ciphertext policy ABE," Proc. ACM Conf. Comput. Commun. Security, pp.456–465, 2007.
[23] V. Goyal, A. Jain, O. Pandey, and A. Sahai, "Bounded ciphertext policy attribute-based encryption," Proc. ICALP, pp.579–591, 2008.
[24] X. Liang, Z. Cao, H. Lin, and D. Xing, "Provably secure and efficient bounded ciphertext policy attribute based encryption," Proc. ASIACCS, pp.343–352, 2009.
[25] M. Chase, "Multi-authority attribute based encryption," Proc. TCC, LNCS 4329, pp.515–534, 2007.
[26] A. Lewko and B. Waters, "Decentralizing attribute-based encryption," Cryptology ePrint Archive: Rep. 2010/351, 2010.
[27] M. Chase and S.S.M. Chow, "Improving privacy and security in multi-authority attribute-based encryption," Proc. ACM Conf. Comput. Commun. Security, pp.121–130, 2009.
[28] H. Lin, Z. Cao, X. Liang, and J. Shao, "Secure threshold multi authority attribute based encryption without a central authority," Proc. INDOCRYPT 2008, LNCS 5365, pp.426–436, 2008.
[29] J. Hur and K. Kang, "Secure Data Retrieval for Decentralized Disruption-Tolerant Military Networks," IEEE/ACM Trans. Netw., vol.22, no.1, pp.16–26, 2014.
[30] Z. Liu, Z. Cao, Q. Huang, D.S. Wong, and T.H. Yuen, "Fully secure multi-authority ciphertext-policy attribute-based encryption without random oracles," Proc. ESORICS 2011, LNCS 6879, pp.278–297, 2011.
[31] A. Boldyreva, V. Goyal, and V. Kumar, "Identity-based encryption with efficient revocation," Proc. ACM Conf. Comput. Commun. Security, pp.417–426, 2008.
[32] N. Chen, M. Gerla, D. Huang, and X. Hong, "Secure, selective group broadcast in vehicular networks using dynamic attribute based encryption," Proc. Ad Hoc Netw. Workshop, pp.1–8, 2010.
[33] M. Pirretti, P. Traynor, P. McDaniel, and B. Waters, "Secure attribute-based systems," Proc. ACM Conf. Comput. Commun. Security, pp.99–112, 2006.
[34] S. Roy and M. Chuah, "Secure data retrieval based on ciphertext policy attribute-based encryption (CP-ABE) system for the DTNs," Lehigh CSE Tech. Rep., 2009.
[35] P. Junod and A. Karlov, "An efficient public-key attribute-based broadcast encryption scheme allowing arbitrary access policies," Proc. DRM'10, pp.13–24, 2010.
[36] L. Ibraimi, M. Petkovic, S. Nikova, P. Hartel, and W. Jonker, "Mediated ciphertext-policy attribute-based encryption and its application," Proc. WISA, LNCS 5932, pp.309–323, 2009.
[37] S. Yu, C. Wang, K. Ren, and W. Lou, "Attribute Based Data Sharing with Attribute Revocation," Proc. 5th ACM Symposium on Information, Computer and Communications Security, pp.261–270, 2010.
[38] D. Huang and M. Verma, "ASPE: Attribute-based secure policy enforcement in vehicular ad hoc networks," Ad Hoc Networks, vol.7, no.8, pp.1526–1535, 2009.

Kenta Nomura received B.E. and M.E. degrees from Kobe University, Japan, in 2015 and 2017, respectively. His current research interests include information security and cryptography.

Masami Mohri received B.E. and M.E. degrees from Ehime University, Japan, in 1993 and 1995 respectively. She received Ph.D degree in Engineering from the University of Tokushima, Japan in 2002. From 1995 to 1998 she was an assistant professor at the Department of Management and Information Science, Kagawa junior college, Japan. From 1998 to 2002 she was a research associate of the Department of Information Science and Intelligent Systems, the University of Tokushima, Japan. From 2003 to 2007 she was a lecturer of the same department. From 2008 to 2017, she was an associate professor at the Information and Multimedia Center, Gifu University, Japan. Since 2017, she has been an associate professor at the Department of Electrical, Electronic and Computer Engineering, Gifu University, Japan. Her research interests are in coding theory, information security and cryptography. She is a member of IEEE.

Yoshiaki Shiraishi received B.E. and M.E. degrees from Ehime University, Japan, and Ph.D degree from the University of Tokushima, Japan, in 1995, 1997, and 2000, respectively. From 2002 to 2006 he was a lecturer at the Department of Informatics, Kindai University, Japan. From 2006 to 2013 he was an associate professor at the Department of Computer Science and Engineering, Nagoya Institute of Technology, Japan. Since 2013, he has been an associate professor at the Department of Electrical and Electronic Engineering, Kobe University, Japan. His current research interests include information security, cryptography, computer network, and knowledge sharing and creation support. He received the SCIS 20th Anniversary Award and the SCIS Paper Award from ISEC group of IEICE in 2003 and 2006, respectively. He received the SIG-ITS Excellent Paper Award from SIG-ITS of IPSJ in 2015. He is a member of IEEE, ACM, and a senior member of IPSJ.

Masakatu Morii received the B.E. degree in electrical engineering and the M.E. degree in electronics engineering from Saga University, Saga, Japan, and the D.E. degree in communication engineering from Osaka University, Osaka, Japan, in 1983, 1985, and 1989, respectively. From 1989 to 1990 he was an Instructor in the Department of Electronics and Information Science, Kyoto Institute of Technology, Japan. From 1990 to 1995 he was an Associate Professor at the Department of Computer Science, Faculty of Engineering, Ehime University, Japan. From 1995 to 2005 he was a Professor at the Department of Intelligent Systems and Information Science, Faculty of Engineering, the University of Tokushima, Japan. Since 2005, he has been a Professor at the Department of Electrical and Electronic Engineering, Faculty of Engineering, Kobe University, Japan. His research interests are in error correcting codes, cryptography, discrete mathematics and computer networks and information security. He is a member of the IEEE.