PAPER *Special Section on Advanced Log Processing and Office Information Systems*

Attribute Revocable Attribute-Based Encryption with Forward Secrecy for Fine-Grained Access Control of Shared Data

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

SUMMARY Ciphertext-Policy Attribute-Based Encryption (CP-ABE) is suitable for data access control on cloud storage systems. In ABE, to revoke users' attributes, it is necessary to make them unable to decrypt ciphertexts. Some CP-ABE schemes for efficient attribute revocation have been proposed. However, they have not been given a formal security proof against a revoked user, that is, whether they satisfy forward secrecy has not been shown or they just do not achieve fine-grained access control of shared data. We propose an attribute revocable attribute-based encryption with the forward secrecy for fine-grained access control of shared data. The proposed scheme can use both "AND" and "OR" policy and is IND-CPA secure under the Decisional Parallel Bilinear Diffie-Hellman Exponent assumption in the standard model.

key words: *cryptographic cloud storage, ciphertext-policy attribute-based encryption, semantic security, attribute revocation, proxy re-encryption*

1. Introduction

Sharing service on a cloud storage has a risk of information leakage caused by unauthorized access and service provider's abuse. In order to prevent the risk, data owners encrypt sharing data on the cloud storage so that only authorized users can decrypt.

Ciphertext-policy attribute-based encryption (CP-ABE) [1], [2] is suitable for data access control of cloud storage systems. In CP-ABE, data owners choose an access structure and encrypt message under the access structure. The set of attributes assigned to users is embedded in his secret key. A user is able to decrypt ciphertexts if his attributes satisfy the access structure of ciphertexts.

In ABE, to revoke users' attributes, it is necessary to make them unable to decrypt ciphertext. In simple user's attribute revocation, when one's attributes are revoked, a data owner re-encrypts shared data so that revoked user cannot decrypt. However, it is not realistic to re-encrypt all shared data.

Some attribute revocable CP-ABE schemes have been proposed [3]–[5]. In these schemes, the authority can revoke the only specified attributes.

In the scheme of [3], the authority can delegate re-encryption and secret key update to a proxy server by proxy re-encryption. When a user accesses encrypted shared data, the proxy server re-encrypts them. Although this scheme has been given a formal security proof against an unauthorized user with semi-trusted proxy servers, it has not been given a formal security proof against a revoked user.

In the scheme of [4], a service provider provides a data outsourcing service. It consists of data servers and a data service manager. Outsourced data from data owners are stored in the data servers. The data server manager distributes key encryption keys (KEKs) to each user. The data service manager re-encrypts a ciphertext by an attribute group key. Then, it encrypts the attribute group key by using KEKs so that authorized users can decrypt. The data service manager is assumed to be honest-but-curious. As the number of system users has increases, the number of KEKs also increases and the management of KEKs becomes more complicated. This scheme also has not been given a formal security proof against a revoked user.

In the scheme of [5], any user cannot decrypt a ciphertext encrypted by a public key the authority generated. When a user downloads encrypted sharing data from the cloud server, the cloud server re-encrypts it by proxy re-encryption so that the user can decrypt it. When revoked user downloads encrypted sharing data, the cloud server does not re-encrypt it, so revoked user cannot decrypt it. This scheme has a limitation in access policy because it only supports "AND" policy.

In this paper, we propose an attribute revocable attribute-based encryption with the forward secrecy for fine-grained access control of shared data. In the context of attribute-based encryption, the forward secrecy means that any user whose attributes were revoked should be prevented from accessing the plaintext of data after revocation, unless other valid attributes he holds satisfy the access policy [4]. In the proposed scheme, it is possible to use not only "AND" but also "OR" for an access policy and the authority can revoke the only specified user's attribute. Furthermore, the proposed scheme meets the following security requirements in [3]–[5] besides the forward secrecy.

1. Data confidentiality: Unauthorized users who do not have enough attributes satisfying the access policy and the cloud server should be prevented from accessing the plaintext of the data.
2. Collusion-resistance: If multiple unauthorized users

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.
[†††]The author is with Nagoya Institute of Technology, Nagoya-shi, 466–8555 Japan.
[*]Presently, with PwC Cyber Services LLC.
a) E-mail: zenmei@port.kobe-u.ac.jp

DOI: 10.1587/transinf.2016OFP0008

and the cloud server collude, when they do not have enough attributes satisfying the access policy, they cannot decrypt a ciphertext.

We define attack model 1 as an attack by unauthorized users and the cloud server. Moreover, we define attack model 2 as an attack by a revoked user. We prove the proposed scheme is IND-CPA secure against attacks defined in each attack model under the Decisional Parallel Bilinear Diffie-Hellman Exponent (DPBDHE) assumption in the standard model.

2. Preliminaries

2.1 Bilinear Maps

Let G_1, G_2 be two cyclic groups of a prime order p. Let g be a generator of G_1. A bilinear map is a map e: $G_1 \times G_1 \rightarrow G_2$ with the following properties:

1. Bilinearity: for all u, $v \in G_1$ and a, $b \in Z_p$, we have $e(u^a, v^b) = e(u, v)^{ab}$
2. Non-degeneracy: $e(g, g) \neq 1$
3. Computability: There is an efficient algorithm to compute $e(u, v)$ for all u, $v \in G_1$

2.2 Decisional Parallel Bilinear Diffie-Hellman Exponent (DPBDHE) Assumption

In [2], the Decisional q-Parallel Bilinear Diffie-Hellman Exponent defined as follows. Choose a group G_1 of prime order p according to the security parameter. Let a, s, $b_1, \dots,$ $b_q \in Z_p$ be chosen at random and $g \in G_1$ be a generator of G_1. If an adversary is given $\vec{y} =$

$$\begin{aligned}&g, g^s, g^a, \dots, g^{a^q}, g^{a^{q+2}}, \dots, g^{a^{2q}}\\ &\forall_{1 \leq j \leq q}\ g^{s \cdot b_j}, g^{a/b_j}, \dots, g^{a^q/b_j},\ , g^{a^{q+2}/b_j}, \dots, g^{a^{2q}/b_j}\\ &\forall_{1 \leq j,k \leq q, k \neq j}\ g^{a \cdot s \cdot b_k/b_j}, \dots, g^{a^q \cdot s \cdot b_k/b_j}\end{aligned}$$

It must remain hard to distinguish $e(g, g)^{a^{q+1}s} \in G_2$ from a random element in $R \in G_2$.

An algorithm $\mathcal{A}$ that outputs $z \in \{0, 1\}$ has advantage ϵ in solving decisional q-parallel BDHE in G_1 if

$$\begin{aligned}&|\Pr[\mathcal{A}(\vec{y}, T = e(g, g)^{a^{q+1}s}) = 0]\\ &-\Pr[A(\vec{y}, T = R) = 0]| \geq \epsilon\end{aligned}$$

We say that the (decision) q parallel-BDHE assumption holds if no polytime algorithm has a non-negligible advantage in solving the decisional q-parallel BDHE problem.

2.3 Access Structure [6]

Let $\{\boldsymbol{P_1}, \boldsymbol{P_2}, \dots, \boldsymbol{P_n}\}$ be the set of parties. A collection $\mathbb{A} \subseteq \boldsymbol{2}^{\{\boldsymbol{P_1}, \boldsymbol{P_2}, \dots, \boldsymbol{P_n}\}}$ is monotone if $\forall \boldsymbol{B}$, $\boldsymbol{C}$: if $\boldsymbol{B} \in \mathbb{A}$ and $\boldsymbol{B} \subseteq \boldsymbol{C}$ then $\boldsymbol{C} \in \mathbb{A}$. An access structure (respectively, monotone access structure) is a collection (respectively, monotone collection) $\mathbb{A}$ of nonempty subsets of $\{\boldsymbol{P_1}, \boldsymbol{P_2}, \dots, \boldsymbol{P_n}\}$, i.e., $\mathbb{A} \subseteq \boldsymbol{2}^{\{\boldsymbol{P_1}, \boldsymbol{P_2}, \dots, \boldsymbol{P_n}\}} \setminus \{\emptyset\}$. The sets in $\mathbb{A}$ are called the authorized sets, and the sets not in $\mathbb{A}$ are called the unauthorized sets.

In our context, the role of the parties is defined by attributes. An access structure $\mathbb{A}$ contains the authorized sets of attributes. Unless otherwise stated, by an access structure we mean a monotone access structure.

2.4 Linear Secret Sharing Schemes (LSSS) Definition 1 (Linear Secret Sharing Schemes (LSSS) [2], [6])

As shown in [6], any LSSS defined as above enjoys the linear reconstruction property defined as follows. A secret-sharing scheme Π over a set of parties $\mathcal{P}$ is called linear (over Z_p) if

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

Suppose that Π is an LSSS for the access structure $\mathbb{A}$. Let $S \in \mathbb{A}$ be any authorized set, and let $I \subset \{1, 2, \dots, l\}$. Then, there exist constants $\{\omega_i \in Z_p\}_{i \in I}$ 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, these constants $\{\omega_i\}$ can be found in polynomial time in the size of the share-generating matrix M [6].

3. System Model and Definition

3.1 Entities

There are four entities in the proposed scheme as follows.

User: Users downloads shared data from the cloud server.

Data Owner: Data owners encrypt shared data then uploads them to the cloud Server.

Authority: The authority manages attributes in the system and publishes the parameter used for encryption. It generates a secret key that user's attributes are embedded in and the re-encryption key used for re-encryption and updates a secret key. The authority is a trusted party.

Cloud Server: The cloud server stores shared data. It re-encrypts encrypted shared data and updates a secret key by using the re-encryption key received from the authority. Similar to the previous scheme [5], we assume cloud servers to be honest-but-curious. That is, they will honestly execute the tasks assigned by legitimate parties in the system. However, they would like to learn information of encrypted shared data as much as possible.

In the previous scheme [4], when an attribute revocation event occurs, a semi-trusted proxy server which is deployed in data server re-encrypts data by using proxy re-key and updates a component of secret key by using it. In the proposed scheme, when a user accesses data after an attribute revocation event, Cloud Server re-encrypts data by using re-encryption key. Both entities do not have secret key, then they cannot decrypt encrypted data. From the above, our storage is similar in trust model to the proxy server in [4], if the proxy server re-encrypts data and updates keys at the timing of user's access after attribute revocation events.

3.2 Algorithm Definition

Our proposed scheme is composed of 5 algorithms: Auth.Setup, Auth.Ext, DO.Enc, C.ReEnc, U.Dec.

-Auth.Setup: The setup algorithm takes as input the security parameter λ which represents encryption strength and attribute universe description U which represents the number of system attributes. It outputs the public parameter PK, the master secret key MK and the re-encryption key RK.

-Auth.Ext: The key extraction algorithm takes as input the master key MK, and a set of attributes S. It outputs the secret key SK.

-DO.Enc: The encryption algorithm takes as input the public parameter PK, an access structure $\mathbb{A}$ and a message $\mathcal{M}$. It outputs the ciphertext CT'.

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

-U.Dec: The decryption algorithm takes as input the secret key SK for a set S and the ciphertext CT for an access structure $\mathbb{A}$. If the set of attributes S satisfies the access structure $\mathbb{A}$, it outputs the message $\mathcal{M}$.

3.3 Security Definitions

We prove that unauthorized users, the cloud server and revoked users cannot decrypt ciphertext encrypted by the proposed scheme. We define the attack model 1 as an attack by unauthorized users and the cloud server. Moreover, we define the attack model 2 as an attack by a revoked user. The cloud server would like to learn information of encrypted shared data as much as possible, but we assume the cloud server is honest, we do not consider active attacks from it by colluding with revoked users as in [3]–[5]. We prove the proposed scheme is IND-CPA secure against attack models and show that the proposed scheme satisfies data confidentiality, collusion-resistance and forward secrecy. In our security model, the adversary will choose to be challenged on an encryption to an access structure $\mathbb{A}^*$ which is called the challenge access structure and can ask for any secret key SK such that SK does not satisfy $\mathbb{A}^*$.

3.3.1 Attack Model 1

In this model, we assume an attack by unauthorized users and the cloud server. Security in this model is defined with the following game.

Init. The adversary A submits the challenge access structure $\mathbb{A}^*$ to the challenger C.

Setup. The challenger C runs setup algorithm and gives the public parameter PK and the re-encryption key RK to the adversary A.

Phase1. The adversary A can issue the following query.
Ext query: The adversary A submits a set of attributes S where S does not satisfy the access structure $\mathbb{A}^*$ to the challenger C. The challenger C gives the secret key SK corresponding to S.

Challenge. The adversary A submits two equal length messages M_0, M_1. The challenger C flips a random coin b. Next, the challenger C encrypts M_b under $\mathbb{A}$ and computes the ciphertext CT'^*. Then, the challenger C runs re-encryption algorithm and gives the re-encryption ciphertext CT^* to the adversary A.

Phase2. Phase1 is repeated.

Guess. The adversary A outputs his guess b' of b.

The advantage of an adversary A in this game is defined as

$$\Pr[b' = b] - \frac{1}{2}.$$

A ciphertext-policy attribute-based encryption scheme is IND-CPA secure in this model if all polynomial time adversaries have at most a negligible advantage in the above game.

3.3.2 Attack Model 2

In this model, we assume an attack by a revoked user. Security in this model is defined with the following game.

Init. The adversary A submits the challenge access structure $\mathbb{A}^*$ and a revoked attribute x^* where $x^* \in \mathbb{A}^*$ to the challenger C.

Setup. The challenger C runs setup algorithm and gives the public parameter PK to the adversary A.

Phase1. The adversary A can issue the following query.
Ext query: The adversary A submits a set of attributes S where S does not satisfy the access structure $\mathbb{A}^*$ and $x^* \in S$ to the challenger C. The challenger C gives the secret key SK corresponding to S.

Challenge. The adversary A submits two equal length messages M_0, M_1. The challenger C flips a random coin b. Next, the challenger C encrypts M_b under $\mathbb{A}$ and computes the ciphertext CT'^*. Then, the challenger C runs re-encryption algorithm and gives the re-encryption ciphertext CT^* to the adversary A.

Phase2. Phase1 is repeated.

Guess. The adversary A outputs his guess b' of b.

The advantage of an adversary A in this game is defined as

$$\Pr[b' = b] - \frac{1}{2}.$$

A ciphertext-policy attribute-based encryption scheme is IND-CPA secure in this model if all polynomial time adversaries have at most a negligible advantage in the above game.

4. Our Scheme

4.1 Overview

The proposed scheme is based on Water's scheme of CP-ABE [2]. It is proved that Waters's scheme is IND-CPA secure under the DBDHE assumption. Moreover, Water's scheme supports any LSSS access structure, so it is possible to use "AND" and "OR" for an access policy.

In the proposed scheme, any user cannot decrypt a ciphertext encrypted by a public key the authority generated. When a user downloads an encrypted sharing data from the cloud server, the server re-encrypts it by proxy re-encryption so that the user can decrypt it. When a revoked user downloads encrypted sharing data, the cloud server doesn't re-encrypt it, so the revoked user cannot decrypt it [5].

4.2 Algorithm

-Auth.Setup: The setup algorithm takes as input the number of system attributes U. It first chooses a group G_1 of prime order p, a generator $g \in G_1$. It then chooses random α, a, $f_1, \ldots, f_U$, $d_1, \ldots, d_U \in Z_p$. The public parameter are $PK := (g, e(g,g)^{\alpha}, g^a, F_1 := g^{f_i}, \ldots, F_U)$. The master key is $MK := \alpha$. The re-encryption key are $RK := (rk_1 := d_1/f_1, \ldots, rk_U := d_U/f_U)$.

-Auth.Ext: The key extraction algorithm takes as input the master key MK, and a set of attributes S. It first chooses a random $t \in Z_p$. It then outputs secret key $SK := (K, L, \{K_x | x \in S\}) = (g^{\alpha} g^{at}, g^t, \{g^{d_x t} | x \in S\})$.

-DO.Enc: The encryption algorithm takes as input the public parameter PK, an LSSS access structure (M, ρ), and a message $\mathcal{M}$. The function ρ associates rows of M to attributes. Let M be an $l \times n$ matrix. It first chooses a random vector $\vec{v} = (s, y_2, \ldots, y_n) \in Z_p$. For $i = 1$ to l, it computes $\lambda_i := \vec{v} \cdot M_i$. It then chooses random $r_1, \ldots, r_l \in Z_p$ and outputs the ciphertext

$$CT' := (C, C', (C_1, D'_1), \ldots, (C_l, D'_l)) = \\ <Ke(g,g)^{\alpha s}, g^s, (g^{a\lambda_1} F_{\rho(1)}^{-r_1}, g^{r_1}), \ldots, (g^{a\lambda_l} F_{\rho(l)}^{-r_l}, g^{r_l})>$$

with (M, ρ).

-C.ReEnc: The re-encryption algorithm takes as input a ciphertext CT', a set of attribute S and re-encryption key RK. Let M be an $l \times n$ matrix. For $i = 1$ to l, if $\rho(i) \in S$, the re-encryption algorithm computes

$$D_i := (D'_i)^{-rk_{\rho(i)}} = g^{f_i r_i / d_i}.$$

If $\rho(i) \notin S$, $D_i := D'_i$. It outputs re-encrypted ciphertext $CT := ((M, \rho), C, C', (C_1, D_1), \ldots, (C_l, D_l))$.

-U.Dec: The decryption algorithm takes as input a secret key SK for a set S and a re-encrypted ciphertext CT for an access structure (M, ρ). Suppose that S satisfies the access structure and let I be defined as $I = \{i : \rho(i) \in S\}$. Then, let $\{\omega_i \in Z_p\}_{i \in I}$ be as set of constants such that if $\{\lambda_i\}$ are valid shares of the secret s according to M, then $\sum_{i \in I} \omega_i \lambda_i = s$. The decryption algorithm first computes

$$\frac{e(C', K)}{\prod_{i \in I} (e(C_i, L) e(D_i, K_{\rho(l)}))^{\omega_i}} = \\ \frac{e(g,g)^{\alpha s} e(g,g)^{ast}}{\prod_{i \in I} e(g,g)^{ta\lambda_i \omega_i}} = e(g,g)^{\alpha s}$$

It can then decrypt the message $\mathcal{M} = C / e(g,g)^{\alpha s}$.
We show the flow of our scheme in Fig. 1.

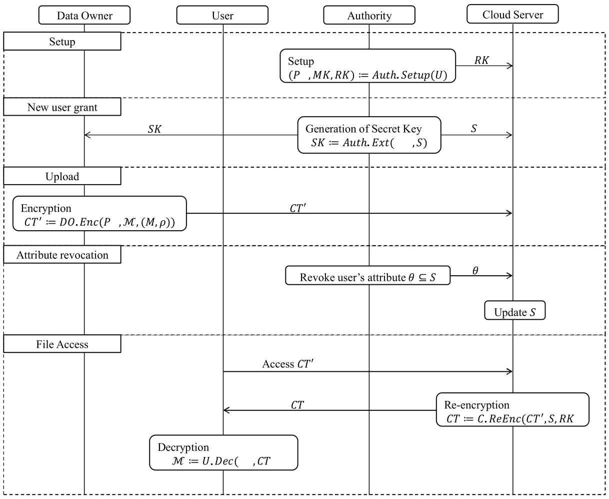
Data Owner
User
Authority
Cloud Server
Setup
Setup
$(P\ \ , MK, RK) := Auth.Setup(U)$
RK
New user grant
SK
Generation of Secret Key
$SK := Auth.Ext(\ \ , S)$
S
Upload
Encryption
$CT' := DO.Enc(P\ \ , \mathcal{M}, (M, \rho))$
CT'
Attribute revocation
Revoke user's attribute $\theta \subseteq S$
θ
Update S
File Access
Access CT'
CT
Re-encryption
$CT := C.ReEnc(CT', S, RK$
Decryption
$\mathcal{M} := U.Dec(\ \ , CT$

Fig. 1 Flow of the proposed scheme

4.3 Attribute Revocation Procedure

The attribute revocation is done by the following procedure:

1. An authority sends the user's revoked attitude $\theta \subseteq S$ to a cloud server.
2. A cloud server updates the attribute set of the user based on the revocation message.
3. When the user accesses the data in the cloud server, the cloud server runs re-encryption algorithm in accordance with an updated attribute set of the user, and sends the re-encrypted ciphertext to the user.

5. Security Proof

We prove that unauthorized users and the cloud server and revoked user cannot decrypt ciphertext CT that was encrypted by using the proposed scheme.

5.1 Security Proof in the Attack Model 1

Theorem 1 Suppose the decisional q-parallel BDHE assumption holds and a challenge matrix of size is $l^* \times n^*$ where $l^* \times n^* \leq q$, our scheme is IND-CPA secure in the attack model 1.

Proof Suppose we have adversary A with non-negligible advantage ϵ against our scheme in the attack model 1. Moreover, suppose it chooses a challenge matrix M^* where both dimensions are at most q. We show how to build a simulator, B, that plays the decisional q-parallel

BDHE problem.

Init. The simulator takes in a q-parallel BDHE challenge $\vec{y}$, T. The adversary gives the simulator B the challenge access structure (M^*, ρ^*), where M^* has n^* columns.

Setup. The simulator B generates the public parameter PK as follows. The simulator B chooses random $\alpha' \in Z_p$ and implicitly sets $\alpha = \alpha' + a^{q+1}$ by letting $e(g,g)^{\alpha} = e(g,g)^{\alpha} = e(g^a, g^{a^q})e(g,g)^{\alpha'}$. For each x for $1 \leq x \leq U$ begin by choosing a random value z_x. Let X denote the set of indices i, such that $\rho^*(i) = x$. The simulator B programs F_x as

$$F_x = g^{z_x} \prod_{i \in X} g^{aM^*_{i,1}/b_i} \cdot g^{a^2 M^*_{i,2}/b_i} \cdots g^{a^{n^*} M^*_{i,n^*}/b_i}$$

Note that if $X = \emptyset$ then we have $F_x = g^{z_x}$. The simulator B gives the adversary A the public parameter $PK := (g, e(g,g)^{\alpha}, g^a, F_1, \ldots, F_U)$.

Phase1. The adversary A issues the following query:

Ext query: The adversary A submits a set of attributes S where S does not satisfy the access structure M^* to the challenger. The simulator first chooses a random $r, rk_1, \ldots, rk_U \in Z_p$. Then it finds a vector $\vec{w} = (w_1, \ldots, w_{n^*}) \in Z_p^{n^*}$ such that $w_1 = -1$ and for all i where $\rho(i) \in S$ we have that $\vec{w} \cdot M^*_i = 0$.
The simulator B begins by implicitly defining t as

$$r + w_1 a^q + w_2 a^{q-1} + \cdots + w_{n^*} a^{q-n^*+1}$$

It performs this by setting $L = g^r \Pi_{i,\ldots,n^*}(g^{(a^{q+1-i})^{w_i}}) = g^t$. The simulator can compute K as

$$K = g^{\alpha'} g^{ar} \prod_{i=2,\ldots,n^*} \left(g^{a^{q+2-i}}\right)^{w_i}$$

The simulator B computes $\{K_x | x \in S\}$ as follows. If there is no i such that $\rho^*(i) = x$, it computes $K_x = L^{z_x rk_x}$. If there is i such that $\rho^*(i) = x$, let X be the set of all i such that $\rho^*(i) = x$. The simulator B computes K'_x as

$$L^{z_x} \prod_{i \in X} \prod_{j=1,\ldots,n^*} \left(g^{(a^j/b_i)r} \prod_{\substack{k=1,\ldots,n^* \\ k \neq j}} \left(g^{(a^{q+1+j-k}/b^i)}\right)^{w_k} \right)^{M^*_{i,j}}$$

and computes $K_x = (K'_x)^{rk_x}$
It gives the adversary A secret key $SK := (K, L, \{K_x | x \in S\})$.

Challenge. The adversary A submits two equal length messages $\mathcal{M}_0$, $\mathcal{M}_1$. The simulator B flips a random coin $b \in \{0,1\}$. It computes $C = \mathcal{M}_b T \cdot e(g^s, g^{\alpha'})$, $C = sP$. It chooses random $y'_2, \ldots, y'_{n^*} \in Z_p$ and the shares of the secret using the vector $\vec{v} = (s, sa + y'_2, sa^2 + y'_3, \ldots, sa^{n-1} + y'_{n^*}) \in Z_p^{n^*}$. In addition, it chooses random values $r'_1, \ldots, r'_l \in Z_p$. For $i = 1, \ldots, n^*$, we define R_i as the set of all $k \neq i$ such that $\rho^*(i) = \rho^*(k)$. The challenge ciphertext components are then generated as

$$D'_i = g^{-r'_i} g^{-sb_i}$$

$$D_i = (D'_i)^{1/rk_{\rho(i)}}$$

$$C_i = h^{r'_i}_{\rho^*(i)} \left(\prod_{j=2,\ldots,n^*} (g^a)^{M^*_{i,j} y'_j} \right) (g^{b_i \cdot s})^{-z_{\rho^*(i)}}$$

$$\left(\prod_{k \in R_i} \prod_{j=1,\ldots,n} \left(g^{a^j \cdot s \cdot (b_i/b_k)}\right)^{M^*_{k,j}} \right)$$

Phase2. Phase 1 is repeated.

Guess. The adversary A will eventually output a guess b' of b. The simulator then outputs 0 to guess that $T = e(g,g)^{a^{q+1}s}$ if $b' = b$; otherwise, it outputs 1 to indicate that it believes T is a random group element $R \in G_2$. When T is a tuple, the simulator B gives a perfect simulation, so we have that

$$\Pr = [\mathrm{B}(\vec{y}, T = e(g,g)^{a^{q+1}s}) = 0] = \frac{1}{2} + \epsilon.$$

When T is a random group element, the message $\mathcal{M}_b$ is completely hidden from the adversary and we have $\Pr = [\mathrm{B}(\vec{y}, T = R) = 0] = \frac{1}{2}$. Therefore, the simulator B can play the decisional q-parallel BDHE game with non-negligible advantage.

5.2 Security Proof in the Attack Model 2

Theorem 1 Suppose the decisional q-parallel BDHE assumption holds and a challenge matrix whose size is $l^* \times n^*$ where $l^* \times n^* \leq q$, our scheme is IND-CPA secure in the attack model 2.

Proof Suppose we have an adversary A with non-negligible advantage ϵ against our scheme in the attack model 2. Moreover, suppose it chooses a challenge matrix M^* where both dimensions are at most q. We show how to build a simulator B that plays the decisional q-parallel BDHE problem.

Init. The simulator takes in a q-parallel BDHE challenge $\vec{y}$, T. The adversary gives the simulator B the challenge access structure (M^*, ρ^*) and x^*, where M^* has n^* columns and there must exist i such that $\rho^*(i) = x^*$.

Setup. The simulator B generates the public parameter PK as follows. The simulator B chooses random $\alpha' \in Z_p$ and implicitly sets $\alpha = \alpha' + a^{q+1}$ by letting $e(g,g)^{\alpha} = e(g,g)^{\alpha} = e(g^a, g^{a^q})e(g,g)^{\alpha'}$. For each $x \neq x^*$ for $1 \leq x \leq U$ begin by choosing random values z_x, $rk_x \in Z_p$. Let X denote the set of indices i, such that $\rho^*(i) = x$. The simulator B programs F_x as

$$F_x = g^{z_x} \prod_{i \in X} g^{aM^*_{i,1}/b_i} \cdot g^{a^2 M^*_{i,2}/b_i} \cdots g^{a^{n^*} M^*_{i,n^*}/b_i}$$

Note that if $X = \emptyset$ then we have $F_x = g^{z_x}$. For the revoked attribute x^*, it chooses a random $z_{x^*} \in Z_p$. Let X^* denote the set of indices i, such that $\rho^*(i) = x^*$. It computes rk_{x^*} as

$$rk_{x^*} = z_{x^*} / (z_{x^*} + (aM^*_{i,1}/b_i) +$$
$$(a^2 M^*_{i,2}/b_i) + \cdots + (a^{n^*} M^*_{i,n^*}/b_i)).$$

The simulator B gives the adversary A the public parameter $PK := (g, e(g,g)^{\alpha}, g^{a}, F_1, \dots, F_U)$.

Phase1. The adversary A issues following query:

Ext query: The adversary A submits a set of attributes S where $S - x^*$ does not satisfy the access structure M^* and $x^* \in S$ to the challenger. Then it finds a vector $\vec{w} = (w_1, \dots, w_{n^*}) \in Z_p^{n^*}$ such that $w_1 = -1$ and for all i where $\rho(i) \in S$ we have that $\vec{w} \cdot M_i^* = 0$.
The simulator B begins by implicitly defining t as

$$r + w_1 a^q + w_2 a^{q-1} + \cdots + w_{n^*} a^{q-n^*+1}$$

It performs this by setting $L = g^r \varPi_{i,\dots,n^*} (g^{(a^{q+1-i})^{w_i}}) = g^t$. The simulator can compute K as

$$K = g^{\alpha'} g^{ar} \prod_{i=2,\dots,n^*} \left(g^{a^{q+2-i}}\right)^{w_i}$$

The simulator B computes $\{K_x | x \in S\}$ as follows. If there is no i such that $\rho^*(i) = x$, it computes $K_x = L^{z_x rk_x}$. If there is i such that $\rho^*(i) = x$, let X be the set of all i such that $\rho^*(i) = x$. The simulator B computes K_x as follows.
If $x = x^*$, it defines K'_{x^*} as

$$L^{z_{x^*}} \prod_{i \in X} \prod_{j=1,\dots,n^*} \left(g^{(a^j/b_i)r} \prod_{\substack{k=1,\dots,n^* \\ k \neq j}} \left(g^{(a^{q+1+j-k}/b^i)}\right)^{w_k} \right)^{M^*_{i,j}}$$

and computes

$$\begin{aligned} K_x &= L^{(1/z_{x^*})} \\ &= L^{rk_{x^*} \cdot (z_{x^*} + (aM^*_{i,1}/b_i) + (a^2 M^*_{i,2}/b_i) + \cdots + (a^{n^*} M^*_{i,n^*}/b_i))} \\ &= (K'_{x^*})^{rk_{x^*}}. \end{aligned}$$

If $x \neq x^*$, it computes K'_x as

$$L^{z_x} \prod_{i \in X} \prod_{j=1,\dots,n^*} \left(g^{(a^j/b_i)r} \prod_{\substack{k=1,\dots,n^* \\ k \neq j}} \left(g^{(a^{q+1+j-k}/b^i)}\right)^{w_k} \right)^{M^*_{i,j}}$$

and $K_x = (K'_x)^{rk_x}$.
It gives the adversary A secret key $SK := (K, L, \{K_x | x \in S\})$.

Challenge. The adversary A submits two equal length messages $\mathcal{M}_0$, $\mathcal{M}_1$. The simulator B flips a random coin $b \in \{0, 1\}$. It computes $C = \mathcal{M}_b T \cdot e(g^s, g^{\alpha'})$, $C = sP$. It chooses random $y'_2, \dots, y'_{n^*} \in Z_p$ and the shares of the secret using the vector $\vec{v} = (s, sa + y'_2, sa^2 + y'_3, \dots, sa^{n-1} + y'_{n^*}) \in Z_p^{n^*}$. In addition, it chooses random values $r'_1, \dots, r'_l \in Z_p$. For $i = 1, \dots, n^*$, we define R_i as the set of all $k \neq i$ such that $\rho^*(i) = \rho^*(k)$. The challenge ciphertext components are then generated as

$$C_i = h_{\rho^*(i)}^{r'_i} \left(\prod_{j=2,\dots,n^*} (g^a)^{M^*_{i,j} y'_j} \right) (g^{b_i \cdot s})^{-z_{\rho^*(i)}}$$

$$\cdot \left(\prod_{k \in R_i} \prod_{j=1,\dots,n} \left(g^{a^j \cdot s \cdot (b_i/b_k)}\right)^{M^*_{k,j}} \right)$$

If $\rho^*(i) = x^*$,

$$D_i = -r'_i T_{\rho^*(i)} - s b_i T_{\rho^*(i)}$$

If $\rho^*(i) \neq x^*$,

$$\begin{aligned} D'_i &= -r'_i T_{\rho^*(i)} - s b_i T_{\rho^*(i)} \\ D_i &= 1/rk_{\rho(i)}(D'_i) \end{aligned}$$

Phase2. Phase 1 is repeated.

Guess. The adversary A will eventually output a guess b' of b. The simulator then outputs 0 to guess that $T = e(g,g)^{a^{q+1}s}$ if $b' = b$; otherwise, it outputs 1 to indicate that it believes T is a random group element $R \in G_2$. When T is a tuple, the simulator B gives a perfect simulation, so we have that

$$\Pr = [\mathrm{B}(\vec{y}, T = e(g,g)^{a^{q+1}s}) = 0] = \frac{1}{2} + \epsilon.$$

When T is a random group element, the message $\mathcal{M}_b$ is completely hidden from the adversary and, we have $\Pr = [\mathrm{B}(\vec{y}, T = R) = 0] = \frac{1}{2}$. Therefore, the simulator B can play the decisional q-parallel BDHE game with non-negligible advantage.

6. Comparisons

In Table 1, we give three comparisons of the proposed scheme with the schemes of [3], [4] and [5]. The first comparison is in terms of the size of the public key (PK), the secret key (SK), the ciphertext (CT), and the re-encryption key (RK). The second comparison is in terms of the computation amount of encryption (Enc), secret key generation (Ext), re-encryption (Re-enc), decryption (Dec), and secret key update (Re-key). The third comparison is in terms of security. As to the size of the public key, the scheme of [4] has the smallest one, followed by the proposed scheme. As for the size of the secret key, the proposed scheme has the smallest one. Both the proposed scheme and the scheme of [4] have equally the smallest size ciphertexts. As to the size of the re-encryption key, if there are users more than the number of attributes, both the proposed scheme and the scheme of [5] have the equally smallest one.

As for the computation amount of encryption and decryption, the proposed scheme and the scheme of [4] have the equally smallest. As to the computation amount of secret key generation, the proposed scheme has the smallest. We consider the case when a secret key is updated under the equal number of the attributes updated by re-encryption. We consider the case that a user access a ciphertext on cloud servers. In schemes of [3] and [4], the secret key is updated when a user accesses a ciphertext on the cloud server. As to the sum of the computation amount of the re-encryption and secret key update computed by the cloud server, [3] and [4] have the same one while [5] and proposed scheme have

Table 1 Comparison of schemes

	The proposed scheme	The scheme of [3]	The scheme of [4]	The scheme of [5]
PK	$(\|U\|+2)$ $\times\|G\|+\|G_T\|$	$(3\|U\|+1)$ $\times\|G\|+\|G_T\|$	$2\times\|G\|+\|G_T\|$	$(3\|U\|+1)$ $\times\|G\|+\|G_T\|$
SK	$(\|S\|+2)\times\|G\|$	$(2\|U\|+1)\times\|G\|$	$(2\|S\|+1)\times\|G\|$ $+\log\|N\|\times\|K\|$	$(2\|U\|+1)\times\|G\|$
CT	$(2\|I\|+1)$ $\times\|G\|+\|G_T\|$	$(\|U\|+1)$ $\times\|G\|+\|G_T\|$	$(2\|I\|+1)$ $\times\|G\|+\|G_T\|$	$(\|U\|+1)$ $\times\|G\|+\|G_T\|$
RK	$\|U\|\times\|Z_p\|$	$2r\|U\|\times\|Z_p\|$	$(2\|N\|-1)\times\|K\|$	$2\|U\|\times\|Z_p\|$
Enc	$(2\|I\|+2)\times exp$	$(\|U\|+2)\times exp$	$2\|I\|+2)\times exp$	$(\|U\|+2)\times exp$
Ext	$(\|S\|+2)\times exp$	$2\|U\|+1)\times *$	$(2\|S\|+2)\times exp$	$2\|U\|+1)\times exp$
Re-enc	$\|R\|\times exp$	$\|R_{CT}\|\times exp$	$\|R_{CT}\|\times exp$	$\|R\|\times ex$
Re-key	None	$\|R_{SK}\|\times exp$	$\|R_{SK}\|\times exp$	None
Dec	$(2\|R\|+1)\times\hat{e}$ $+(2\|R\|+2)\times exp$	$(\|U\|+1)\times\hat{e}$ $+(\|U\|+1)\times exp$	$(2\|R\|+1)\times\hat{e}$ $+(2\|R\|+2)\times exp$	$(\|U\|+1)\times\hat{e}$ $+(\|U\|+1)\times exp$
Security against attack model 1 (5.1)	IND-CPA secure (dqPBDHE assumption)	IND-CPA secure (DBDH assumption)	—	IND-CPA secure (DBDH assumption)
Security against attack model 2 (5.2)	IND-CPA secure (dqPBDHE assumption)	—	—	IND-CPA secure (DBDH assumption)
Access policy	AND, OR	AND	AND, OR	AND

exp: exponentiation in G, $\hat{e}$: bilinear pairing, $|U|$: the number of attributes defined in the system,
$|S|$: the number of attributes in user's key, $|R|$: the number of user's attributes satisfying an access structure,
r: the number of times the attribute revocation event occurs,
$|R_{SK}|$: the number of updated attributes (secret key), $|R_{CT}|$: the number of updated attributes (ciphertext),
$|N|$: the number of total users, $|I|$: the number of attributes in the access structure,
$|K|$: size of the common key

smaller one. That is, when attribute revocation occurs frequently, as to the amount of computation of the cloud server, the proposed scheme and [5] have the equally smallest one.

As for the security, the proposed scheme and the scheme of [5] have been shown that they are IND-CPA secure against an attack by unauthorized users, the cloud server, and a revoked user.

7. Conclusion

This paper proposed an attribute revocable attribute-based encryption with the forward secrecy for fine-grained access control of shared data. In the proposed scheme, it is possible to use not only "AND" but also "OR" for an access policy and the authority can revoke the only specified user's attributes. We proved the proposed scheme is IND-CPA secure against attacks defined as attack models under the Decisional Parallel Bilinear Diffie-Hellman Exponent (DPBDHE) assumption in the standard model, and show that the proposed scheme satisfies data confidentiality, collusion-resistance and forward secrecy.

Our future direction is to implement the proposed scheme and confirm its feasibility.

Acknowledgments

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

References

[1] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-Policy Attribute-Based Encryption," Proc. 2007 IEEE Symposium on Security and Privacy, pp.321–334, 2007.

[2] B. Waters, "Ciphertext-Policy Attribute-Based Encryption: An Expressive, Efficient, and Provably Secure Realization," Proc. 14th International Conference on Practice and Theory in Public Key Cryptography, pp.53–70, 2011.

[3] 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.

[4] J. Hur and D.K. Noh, "Attribute-Based Access Control with Efficient Revocation in Data Outsourcing Systems," IEEE Trans. Parallel Distrib. Syst., vol.22, no.7, pp.1214–1221, 2011.

[5] T. Naruse, M. Mohri, and Y. Shiraishi, "Attribute Revocable Attribute-Based Encryption with Forward Secrecy," IPSJ Journal, vol.55, no.10, pp.2256–2264, 2014. (in Japanese).

[6] A. Beimel, "Secure Schemes for Secret Sharing and Key Distributetion," PhD thesis, Israel Institute of Technology, Technion, Haifa, Israel, 1996.

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.

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.

Takeru Naruse received B.E. and M.E. degrees from Nagoya Institute of Technology, Japan, in 2013 and 2015, respectively. His research interests include information security and cryptography. He received DICOMO2013 symposium Paper Award and Presentation Award in 2013. He is a 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.