Hindawi
Security and Communication Networks
Volume 2020, Article ID 8856592, 11 pages
https://doi.org/10.1155/2020/8856592

Research Article

A New User Revocable Ciphertext-Policy Attribute-Based Encryption with Ciphertext Update

Zhe Liu,[1] **Fuqun Wang**,[1,2] **Kefei Chen**,[1,2] **and Fei Tang**[3]

[1]*Department of Mathematics, Hangzhou Normal University, Hangzhou, China*
[2]*Westone Cryptologic Research Center, Beijing, China*
[3]*School of Cyber Security and Information Law, Chongqing University of Posts and Telecommunications, Chongqing, China*

Correspondence should be addressed to Fuqun Wang; fqwang@hznu.edu.cn

Received 29 August 2020; Revised 30 October 2020; Accepted 22 November 2020; Published 22 December 2020

Academic Editor: Kaitai Liang

The revocable ciphertext-policy attribute-based encryption (R-CP-ABE) is an extension of ciphertext-policy attribute-based encryption (CP-ABE), which can realize user direct revocation and maintain a short revocation list. However, the revoked users can still decrypt the previously authorized encrypted data with their old key. The R-CP-ABE scheme should provide a mechanism to protect the encrypted data confidentiality by disqualifying the revoked users from accessing the previously encrypted data. Motivated by practical needs, we propose a new user R-CP-ABE scheme that simultaneously supports user direct revocation, short revocation list, and ciphertext update by incorporating the identity-based and time-based revocable technique. The scheme provides a strongly selective security proof under the modified decisional q-parallel bilinear Diffie–Hellman Exponent problem, where "strongly" means that the adversary can query the secret key of a user whose attribute set satisfies the challenge ciphertext access structure and whose identity is in the revocation list.

1. Introduction

As a special kind of public key encryption (PKE), attribute-based encryption (ABE) is a one-to-many cryptographic primitive that can offer a fine-grained access control. In general, there are two types of ABE schemes, key-policy attribute-based encryption (KP-ABE) [1–4] and ciphertext-policy attribute-based encryption (CP-ABE) [5–8]. In the KP-ABE scheme, secret key is associated with an access structure, and ciphertext is labeled with a set of attributes. While in the CP-ABE scheme, secret key is related to a set of attributes, and ciphertext is associated with an access structure. Compared with the traditional method of access control system, ABE has many advantages so that it satisfies many applications for network such as cloud storage systems [9–12] and medical e-healthcare systems [13–18].

However, providing an efficient and practical revocation mechanism is very important in ABE since it can prevent a user from accessing encrypted data in cryptosystems by revoking the access authority. There are mainly two methods to revoke users in ABE, namely, direct revocation and indirect revocation. The indirect revocation [19, 20] requires an authority to update key only for the nonrevoked users so that they can continue to decrypt the encrypted data. The revoked users cannot decrypt any newly generated ciphertext since their keys were not updated. However, we cannot implement user instant revocation by using this approach. Suppose an employee's access to the encrypted data is revoked some day before the key update time, he could still decrypt any newly generated encrypted data until the key is updated. If we update the key as soon as a user is revoked to realize user instant revocation, it will be a bottleneck and not practical for a large organization where there may be an army of revoked users. Moreover, the revoked users can still have access to the previously generated encrypted data. The direct revocation [21, 22] allows a public revocation list to be specified directly during encryption so that the ciphertext cannot be decrypted by those users who are in the revocation list even if their attributes/policies satisfy the policies/attributes related to the ciphertext. Ciphertext can only be decrypted by users who are not in the revocation list and whose attributes satisfy the access policy. This method can implement user instant revocation and does not need to update the secret key, while the disadvantage is that the revocation list gets longer over time. It will be inefficient for encryption and decryption, especially for a large system.

1.1. Related Work. Many schemes [23–28] are presented to deal with the revocation in attribute-based access control. Boldyreva et al. in [19] proposed a revocable KP-ABE. In their scheme, the authority stores a revocation list and executes key update algorithm for the nonrevoked users who are not in the revocation list. Using the key update approach, Yu et al. in [25] put forward a revocable CP-ABE. The revoked users cannot decrypt the updated ciphertext, but access policies rarely support logical AND in their contribution. In 2012, Sahai et al. in [20] proposed a concept of revocable-storage ABE. In the scheme, they added a ciphertext delegation and ciphertext updating algorithm so that ciphertext can be decrypted only if the encryption time $t < t'$, where t' is the key expiry time. In detail, the third party server can update stored ciphertext without any interaction with data owners as long as the revocation event happens and the re-encrypted ciphertext cannot be recovered by the revoked users any longer. Using the direct revocation, Balu et al. in [26] put forward a revocable CP-ABE. Their model, however, is weak that the adversary can only query the secret key of a user whose attribute set does not satisfy the challenge ciphertext access policy and whose identity is not in the revocation list.

Wang et al. in [23] proposed a new revocable CP-ABE that incorporates ID-based revocation ability. In their security definition, the adversary can query the secret key of a user whose attribute set satisfies the challenge ciphertext access structure and whose identity is in the revocation list. Nevertheless, the size of the ciphertext is linear with the number of users in the revocation list, which gets longer as time goes by. Liu et al. in [29] proposed a revocable CP-ABE by using direct approach. They put forward a secret key time validation technique to address the issue of growth of the revocation list. Users can decrypt the ciphertext if and only if the validity time period of the secret key completely covers the validity time period of the ciphertext. The size of the ciphertext is only related to the embedded policy, while the size of the secret key is not only linear with the maximum length of the revocation list but also the number of attributes of the user. Their scheme can implement user direct revocation and maintain a short revocation list. However, the revoked users can still decrypt the previously authorized ciphertext with their old key. We take this issue into account where users' access authority changes with time and ciphertext is stored by a third party.

1.2. Our Contribution. We propose a R-CP-ABE scheme that can implement user direct revocation, maintain a short revocation list, and update ciphertext by incorporating the identity-based and time-based revocable technique. The main contributions of this paper can be summarised as follows:

(1) *User direct revocation.* We have a public revocation list that contains the identity of a user who is revoked before the intended expiry time. This revocation list is embedded into the ciphertext by the encryptor to achieve user direct revocation. Users in the revocation list cannot decrypt any newly generated ciphertext even if their attribute set satisfies the access policy.

(2) *Short revocation list.* Once the validity time expires, the users' keys become invalid as they are unable to decrypt any newly generated ciphertext. The revoked users whose keys are expired can be removed from the revocation list after the expiry date of their keys. Therefore, we can maintain a short revocation list.

(3) *Ciphertext update.* In the scheme, the ciphertext can be updated periodically using only publicly available information, and after the update process, all stored encrypted data (no matter how old) become inaccessible to the revoked users.

(4) *Strongly selective security.* Our scheme provides a strongly selective security proof under the modified decisional q-parallel bilinear Diffie–Hellman Exponent problem, where "strongly" means that the adversary can query the secret key of a user whose attribute set satisfies the challenge ciphertext access structure and whose identity is in the revocation list.

2. Preliminaries

2.1. Bilinear Pairings. Let $\mathbb{G}_1$ and $\mathbb{G}_2$ be two cyclic multiplicative groups of prime order p, and g be a generator of $\mathbb{G}_1$. A bilinear map is a function e: $\mathbb{G}_1 \times \mathbb{G}_1 \longrightarrow \mathbb{G}_2$ with the following properties:

(i) Bilinearity: $e(aP, bQ) = e(P, Q)^{ab}$ for all $P, Q \in \mathbb{G}_1$ and all $a, b \in \mathbb{Z}_p$

(ii) Nondegeneracy: $e(g, g) \neq 1$

(iii) Computability: there is a polynomial time algorithm to compute $e(P, Q)$ for any $P, Q \in \mathbb{G}_1$

2.2. Access Structure. Let a set of parties be $\{P_1, P_2, \ldots, P_n\}$. A collection $\mathbb{A} \subseteq 2^{\{P_1, P_2, \ldots, P_n\}}$ is monotone if $\forall B, C$. If $B \in \mathbb{A}$ and $B \subseteq C$, then $C \in \mathbb{A}$. An access structure $\mathbb{A}$ is a collection of nonempty subsets of $\{P_1, P_2, \ldots, P_n\}$, i.e., $\mathbb{A} \subseteq 2^{\{P_1, P_2, \ldots, P_n\}} \backslash \{\varnothing\}$. For an access structure $\mathbb{A}$, the sets in $\mathbb{A}$ are defined as authorized sets. Otherwise, the sets are defined as unauthorized sets.

2.3. Linear Secret-Sharing Schemes (LSSS). An LSSS can represent an access control policy (M, ρ), where M with l rows and n columns is called the share-generating matrix and the function ρ defines the party labeling row i as $\rho(i)$ for all $i = 1, \ldots, l$. A secret-sharing scheme Π over a set of parties is linear over $\mathbb{Z}_p$ if satisfies the following two conditions:

(i) The shares of each parties form a vector over $\mathbb{Z}_p$.

(ii) The column vector $\mathbf{v} = (s, r_2, r_3, \ldots, r_n)$ is the secret to be shared, where $s \in \mathbb{Z}_p$ and $r_2, r_3, \ldots, r_n \in \mathbb{Z}_p$ are chosen randomly. According to Π, M_v is the vector of l shares of the secret s and the share $(M_v)_i$ belongs to party $\rho(i)$.

Our definition is adopted from [30], and it showed that every linear secret-sharing scheme enjoys the linear reconstruction property:

(i) Suppose that Π is an LSSS for the access structure $\mathbb{A}$. Let any authorized set $S \in \mathbb{A}$ and $I = \{i\colon \rho(i) \in S\} \subset \{1, \ldots, l\}$. Then, there exist constants $\{\omega_i \in \mathbb{Z}_p\}_{i \in I}$ such that $\sum_{i \in I} \omega_i \lambda_i = s$ for valid shares $\{\lambda_i\}$ of any secret s, and we can find these constants ω_i in polynomial time.

We use the convention that the vector $(1, 0, 0, \ldots, 0)$ is the target vector for any linear secret-sharing scheme. The target vector $(1, 0, 0, \ldots, 0)$ is in the span of I for any satisfying set of rows I in M. For any unauthorized set of rows I, the target vector is not in the span of I. A vector $\mathbf{w}$ exists such that $\mathbf{w} \cdot (1, 0, 0, \ldots, 0) = -1$.

2.4. Security Assumption. The modified decisional q-parallel bilinear Diffie–Hellman Exponent problem (M-q-parallel-BDHE) is defined as follows. Given

$$\begin{aligned}\mathbf{y} = \Big\{ & g, g^s, g^a, \ldots, g^{(a^q)}, g^{(a^{q+2})}, \ldots, g^{(a^{2q})}, \\ & \forall_{1 \le j \le q}\, g^{a/b_j}, \ldots, g^{a^q/b_j}, g^{a^{q+2}/b_j}, \ldots, g^{a^{2q}/b_j}, \\ & \forall_{1 \le j \le q}\, g^{a \cdot s/b_j}, \ldots, g^{a^q \cdot s/b_j} \Big\},\end{aligned} \tag{1}$$

where $\mathbb{G}_1$ is a group of prime order p with a random generator g and the random exponents $a, s, b_1, b_2, \ldots, b_q \in \mathbb{Z}_p$, in order to distinguish $e(g, g)^{a^{q+1}s} \in \mathbb{G}_2$ from a random element $R \in \mathbb{G}_2$.

The advantage of solving the M-q-parallel-BDHE problem in $\mathbb{G}_1$ with algorithm $\mathscr{B}$ is ε if the following equation holds:

$$\left| \Pr\left[\mathscr{B}\left(\mathbf{y}, T = e(g, g)^{a^{q+1}s} \right) = 0 \right] - \Pr[\mathscr{B}(\mathbf{y}, T = R) = 0] \right| \ge \varepsilon. \tag{2}$$

The M-q-parallel-BDHE assumption holds if the advantage ε of any probabilistic polynomial time (PPT) algorithm to solve the M-q-parallel-BDHE problem is a negligible function of the security parameter.

3. Definition

3.1. Time Period. Similar to the definition of time period in [29], our time period is hierarchical that we use a hierarchical tree to represent the time period for year, month, and day. Let $\mathtt{T}$ be the depth of the hierarchical tree, the first level represents the year, the second level represents the month, and the third level represents the day. Every node has z children, and each node (except the root node) represents a time period in the tree. We assume that all users agree on how to divide time and how to specify each time period. A time period $\tau = (\tau_1, \tau_2, \ldots, \tau_k)$, where the j-th component corresponds to the time period at level j. For example, we use 2020.08.22 to represent a day, 2020.08 to represent a month and so on.

A secret key validity time for a user is a time period from a starting date to an ending date. For example, a user joins the organization on 2019.12.30 and ends on 2020.12.31, and then his secret key validity time is from 2019.12.30 to 2020.12.31. A decryptable time period is a time period set by the encryptor so that only users with validity time completely covers the period can decrypt. For example, suppose the decryptable time period is 2019.12 and the secret key validity of a user is only limited to 2019.12.31. This secret key is unable to decrypt as it does not have a complete cover for the decryptable time period. However, if the decryptable time period is 2019.12.31 and the secret key validity of a user is 2019.12, then it is able to decrypt as it has a complete cover for the decryptable time period.

3.2. Algorithms of R-CP-ABE. The R-CP-ABE scheme consists of five PPT algorithms: Setup, KeyGen, Encrypt, Decrypt, and CTUpdate:

(i) Setup$(\lambda, U, \mathtt{T}, I)$: the setup algorithm takes as input the security parameter λ, the number of attributes in the system U, the depth of the time tree $\mathtt{T}$, and the identity set I. It outputs the public parameters PK and a master key MK.

(ii) KeyGen$(\mathrm{MK}, \mathrm{ID}, S, T)$: the key generation algorithm takes as input the master key MK, a user's ID, a set of attributes S, and a range of time periods T for the user's ID. It outputs a private key $\mathrm{SK}_{(\mathrm{ID}, S, T)}$.

(iii) Encrypt$(\mathrm{PK}, \mathscr{M}, T_c, \mathscr{R}, \mathbb{A} = (M, \rho))$: the encryption algorithm takes as input the public parameters PK, a message $\mathscr{M}$, a decryptable time period T_c, a revoked set $\mathscr{R}$, and an access structure $\mathbb{A}$ over the universe of attributes. It outputs a ciphertext CT.

(iv) Decrypt$(\mathrm{CT}, \mathscr{R}, \mathbb{A}, T_c, \mathrm{SK}_{(\mathrm{ID}, S, T)})$: the decryption algorithm takes as input a ciphertext CT, with a description of a revoked set $\mathscr{R}$, an access structure $\mathbb{A}$ and the time periods T_c, and a private key $\mathrm{SK}_{(\mathrm{ID}, S, T)}$. If and only if the user's identity ID is not in the revocation list, the set S of attributes satisfies the access structure $\mathbb{A}$ associated with CT and the range of validity time periods T completely covering the decryptable time periods T_c, and then it outputs the message $\mathscr{M}$.

(v) CTUpdate$(\mathrm{PK}, \mathrm{CT}, \mathscr{R}', T_c, \mathbb{A} = (M, \rho))$: the ciphertext update algorithm takes as input the public parameters PK, the ciphertext CT, a new revoked set $\mathscr{R}'$, the decryptable time period T_c, and an access structure $\mathbb{A} = (M, \rho)$. It outputs a new ciphertext CT'.

Note that compared to the algorithms of R-CP-ABE scheme [29], we add a ciphertext update algorithm to prevent the revoked users from accessing the previously authorized encrypted data. We do not explicitly propose a key update algorithm as its function can be covered by the

KeyGen algorithm. We run the KeyGen algorithm to generate a new secret key with a new time period for the nonrevoked users during a reasonable period (e.g., employees that renew their contracts when they expire).

3.3. Security Model. Due to the updated ciphertext has the same distribution as the original ciphertext, we only consider the security of the original ciphertext. The security model is described by the following a game between a challenger $\mathscr{C}$ and an adversary $\mathscr{A}$. In the game, $\mathscr{A}$ needs to submit an access structure $\mathbb{A}^*$, a revocation list $\mathscr{R}^*$, and a decryptable time period T_c^* to $\mathscr{C}$ before seeing the public parameters PK. $\mathscr{A}$ can query any private key at any time that cannot be used to decrypt the challenge ciphertext, which derives from the security definitions for identity-based revocation framework in [31] and general CP-ABE systems in [7]. In the security definition, we consider a strong adversary who can query the secret key of a user whose attribute set satisfies the challenge ciphertext access structure and whose identity is in the revocation list.

- (i) Init: the adversary $\mathscr{A}$ submits the challenge access structure $\mathbb{A}*$, the challenge revocation list $\mathscr{R}^*$, and the challenge decryptable time period T_c^* to the challenger $\mathscr{C}$.
- (ii) Setup: $\mathscr{C}$ launches the Setup algorithm to generate the system parameters. It keeps the master key MK and sends the public parameters PK to $\mathscr{A}$.
- (iii) Phase1: $\mathscr{A}$ makes private key queries repeatedly corresponding to the identity ID, the attribute set S, and the range of time periods T such that, for any single returned secret key $\mathrm{SK}_{(\mathrm{ID},S,T)}$, at least one of the following requirements is satisfied:
    - (i) S satisfies the access structure $\mathbb{A}^*$ and the corresponding identity $\mathrm{ID} \in \mathscr{R}^*$
    - (ii) T_c^* is not completely covered in T
- (iv) Challenge: $\mathscr{A}$ submits two equal length messages $\mathscr{M}_0$ and $\mathscr{M}_1$ to $\mathscr{C}$. And then, $\mathscr{C}$ flips a random coin $b \in \{0, 1\}$ and encrypts $\mathscr{M}_b$ under the access structure $\mathbb{A}^*$, the revoked set $\mathscr{R}^*$, and the time period T_c^* to obtain a ciphertext CT $*$. Finally, $\mathscr{C}$ sends the ciphertext CT^* to $\mathscr{A}$.
- (v) Phase2: this phase is completely same as the Phase 1.
- (vi) Guess: $\mathscr{A}$ outputs a guess b' of b.

The advantage of $\mathscr{A}$ winning the game is defined as $\mathrm{Adv}_{\mathscr{A}} = |\Pr[b' = b] - 1/2|$.

- (i) Definition: if no adversary has a nonnegligible advantage to win the above game in polynomial time, then the revocable ciphertext-policy attribute-based encryption scheme is secure.

4. Our Scheme

4.1. Overview. Based on the scheme [23] and the secret key time validation technique in [29], we propose the R-CP-ABE scheme with ciphertext update. We incorporate identity and time period to the generating process of the secret key. The size of the revocation list can be reduced by incorporating validity time period technique. The identity of a user who is revoked before his intended expiry date is embedded into the revocation list by the encryptor to realize user direct revocation. Users in the revocation list cannot decrypt any newly encrypted data. In order to disqualify the revoked users from accessing the previously encrypted data, we provide a ciphertext update mechanism. Finally, our scheme can implement user direct revocation, maintain a short revocation list, and update ciphertext.

4.2. Technique Construction. Similar to the validity time technique in [29] from the hierarchical IBE (HIBE) scheme [13], we represent time period by using a hierarchical tree, which can shorten the size of the secret key. In this hierarchical tree, each node has a corresponding time period associated with the secret key, and the secret key of any node can derive the secret key for children of that node. For example, a user with secret key validity time period for the whole year can derive the key with validity time period for the underlying months of that year.

We select the minimum number of nodes that can represent all the validity time periods by using the set-cover approach. Suppose a user joins the organization on 2019.12.30 and ends on 2020.12.31, then his secret key validity time is from 2019.12.30 to 2020.12.31. He should obtain secret key from the nodes of 2019.12.30, 2019.12.31, and 2020 by using the set-cover approach. Then, the secret time period is the set {(2019.12.30), (2019.12.31), (2020)}.

The detailed construction of the scheme is as follows:

- (i) Setup (U, T, I): U is the number of attributes in the system. The time periods are represented as a z-ary string $\{1, z\}^{\mathrm{T}-1}$. I is the identity set. The algorithm chooses a bilinear group $\mathbb{G}_1$ of prime order p with a random generator g and U random group elements $h_1, h_2, \ldots, h_U \in \mathbb{G}_1$. It also randomly chooses $\alpha, b \in \mathbb{Z}_p$ and $V_0, V_1, \ldots, V_{\mathrm{T}} \in \mathbb{G}_1$. It outputs:

$$\mathrm{PK} = \left\{g, g^b, g^{b^2}, e(g,g)^\alpha, h_1^b, \ldots, h_U^b, V_0, V_1, \ldots, V_{\mathrm{T}}\right\}, \quad (3)$$

  and $\mathrm{MK} = \{\alpha, b\}$.

- (ii) KeyGen $(\mathrm{MK}, \mathrm{ID}, S, T)$: S is the set of attributes of a user with identity $\mathrm{ID} \in I$. T is the time period for the user ID. $\mathbb{T}$ is denoted as the set-cover to represent T which consists of some time elements $\tau = (\tau_1, \tau_2, \ldots, \tau_{k_\tau}) \in \{1, z\}^{k_\tau}$ for any $\tau \in \mathbb{T}$. The algorithm randomly chooses t, $v_\tau \in \mathbb{Z}_p$ for any $\tau \in \mathbb{T}$ and computes

$$D_0 = g^t, \left\{D_{0,\tau} = g^{v_\tau}\right\}_{\tau\in\mathbb{T}}, \left\{D_{1,\tau} = g^\alpha g^{b^2 t}\left(V_0 \prod_{j=1}^{k_\tau} V_j^{\tau_j}\right)^{v_\tau}\right\}_{\tau\in\mathbb{T}}, \left\{L_{j,\tau} = V_j^{\tau_j}\right\}_{j=k_{\tau+1,\ldots \mathrm{T},\tau}\in\mathbb{T}}, \left\{K_x = \left(g^{b\cdot\mathrm{ID}} h_x\right)^{-t}\right\}_{x\in S}. \quad (4)$$

Then, the secret key is

$$\mathrm{SK}_{(\mathrm{ID},S,T)} = \Bigl\{D_0, \bigl\{D_{0,\tau}, D_{1,\tau}, L_{k_\tau+1,\tau}, \ldots, L_{T,\tau}\bigr\}_{\tau\in\mathbb{T}}, \\ \{K_x\}_{x\in S}\Bigr\}. \tag{5}$$

(iii) Encrypt$(\mathrm{PK}, \mathscr{M}, T_c, \mathscr{R}, \mathbb{A} = (M,\rho))$: the revocation list $\mathscr{R} = (\mathrm{ID}_1, \ldots, \mathrm{ID}_r)$ with r revoked users. The message $\mathscr{M} \in \mathbb{G}_2$ and the decryptable time period of the ciphertext is T_c. $\tau_c = (\tau_1, \tau_2, \ldots, \tau_k) \in \{1,z\}^k$ denotes the representation of T_c. It takes as input an LSSS access structure $\mathbb{A} = (M,\rho)$, where M is an $l \times n$ matrix and ρ is a function maps rows of M into attributes. The encryption algorithm chooses a random vector $\mathbf{v} = (s, y_2, \ldots, y_n) \in \mathbb{Z}_p^n$ to share the encryption exponent s. For $i = 1, \ldots, l$, it calculates $\lambda_i = \mathbf{v} \cdot M_i$, where the vector M_i corresponds to the i-th row of M. Let ID_j denote the j-th identity in $\mathscr{R}$. The algorithm also chooses random $\mu_1, \ldots, \mu_r \in \mathbb{Z}_p$ such that $\mu = \mu_1 + \mu_2 + \cdots + \mu_r$. It computes

$$C_0 = \mathscr{M} \cdot e(g,g)^{\alpha s\mu}, C_0' = g^{s\mu}, C_0'' = \left(V_0 \prod_{j=1}^{k} V_j^{\tau_j}\right)^{s\mu}, \\ C_{i,j} = g^{b\lambda_i\mu_j}, C_{i,j}' = \left(g^{b^2 \mathrm{ID}_j} h_{\rho(i)}^{b}\right)^{\lambda_i\mu_j}. \tag{6}$$

Then, $\mathrm{CT} = \bigl\{C_0, C_0', C_0'', C_{i,j}, C_{i,j}'\bigr\}$ along with a description of the revoked set $\mathscr{R}$, the access structure $\mathbb{A} = M, \rho$, and the time periods T_c.

(iv) Decrypt$(\mathrm{CT}, \mathscr{R}, \mathrm{SK}_{(\mathrm{ID},S,T)})$: the decryption algorithm takes as input a ciphertext CT with access structure (M,ρ), the revocation list $\mathscr{R}$, and the private key $\mathrm{SK}_{(\mathrm{ID},S,T)}$. If the following requirements occurs, output $\perp$:

(i) S satisfies the access structure $\mathbb{A}$ and the corresponding identity $\mathrm{ID} \in \mathscr{R}$

(ii) T_c is not completely covered in T, that is, τ_c and all its prefixes are not in $\mathbb{T}$

Otherwise, we have $\mathrm{ID} \notin \mathscr{R}$, and S satisfies the access structure $\mathbb{A} = (M,\rho)$. Define $I = \{i\colon \rho(i) \in S\} \in \{1, 2, \ldots, l\}$. There exists a set of constants $\{\omega_i \in \mathbb{Z}_p\}_{i\in I}$ such that $\sum_{i\in I}\omega_i\lambda_i = s$, if $\{\lambda_i\}$ are valid shares of any secret s according to M. It computes

$$\begin{aligned}
&\prod_{i\in I}\left(\prod_{j=1}^{r}\left[e(C_{i,j}', D_0)\cdot e(C_{i,j}, K_{\rho(i)})\right]^{(1/\mathrm{ID}-\mathrm{ID}_j)}\right)^{\omega_i} \\
&= \prod_{i\in I}\left(\prod_{j=1}^{r}\Bigl[e\Bigl(\Bigl(g^{b^2\lambda_i \mathrm{ID}_j} h_{\rho(i)}^{\lambda_i b}\Bigr)^{\mu_j}, g^{t}\cdot e\bigl(g^{b\lambda_i\mu_j}, \bigl(g^{b\cdot \mathrm{ID}} h_{\rho(i)}\bigr)^{-t}\bigr)\Bigr]^{(1/\mathrm{ID}-\mathrm{ID}_j)}\right)^{\omega_i} \\
&= \prod_{i\in I}\left(\prod_{j=1}^{r}\left[e(g^{b^2\lambda_i \mathrm{ID}_j\cdot\mu_j}, g^{t})\cdot e(g^{b^2\lambda_i \mathrm{ID}\cdot\mu_j}, g^{(-t)})\right]^{(1/\mathrm{ID}-\mathrm{ID}_j)}\right)^{\omega_i} \\
&= \prod_{i\,\in\, I}\left(\prod_{j=1}^{r}\left[e(g,g)^{b^2 t\lambda_i\mu_j(\mathrm{ID}_j-\mathrm{ID})}\right]^{(1/\mathrm{ID}-\mathrm{ID}_j)}\right)^{\omega_i} \\
&= \prod_{i\in I}\left(\prod_{j=1}^{r} e(g,g)^{b^2 t\lambda_i\mu_j}\right)^{-\omega_i} \\
&= \prod_{i\in I}\left(e(g,g)^{\left(\sum_{j=1}^{r}\mu_j\right)b^2 t\lambda_i}\right)^{-\omega_i} \\
&= e(g,tg)^{-b^2 t\mu\sum_{i\in I}\lambda_i\omega_i} \\
&= e(g,tg)^{-b^2 t\mu s}.
\end{aligned} \tag{7}$$

Denote $A = e(D_{1,\tau}, C_0')$. Finally, it computes

$$\frac{C_0 \cdot e\left(D_{0,\tau}, C_0''\right)}{A \cdot \prod_{i\in I}\left(\prod_{j=1}^{r}\left[e\left(C_{i,j}', D_0\right) \cdot e\left(C_{i,j}, K_{\rho(i)}\right)\right]^{\left(1/\mathrm{ID}-\mathrm{ID}_j\right)}\right)^{\omega_i}}. \tag{8}$$

The process is as follows:

$$\begin{aligned}&\frac{C_0 \cdot e\left(D_{0,\tau}, C_0''\right)}{A \cdot \prod_{i\in I}\left(\prod_{j=1}^{r}\left[e\left(C_{i,j}', D_0\right) \cdot e\left(C_{i,j}, K_{\rho(i)}\right)\right]^{\left(1/\mathrm{ID}-\mathrm{ID}_j\right)}\right)^{\omega_i}} \\ &= \frac{\mathscr{M} \cdot e(g,g)^{\alpha s\mu} \cdot e\left(g^{v_\tau}, \left(V_0 \prod_{j=1}^{k} V_j^{\tau_j}\right)^{s\mu}\right)}{e\left(g^{\alpha} g^{b^2 t}\left(V_0 \prod_{j=1}^{k} V_j^{\tau_j}\right)^{v_\tau}, g^{s\mu}\right) \cdot e(g,g)^{-b^2 t\mu s}} \\ &= \frac{\mathscr{M} \cdot e(g,g)^{\alpha s\mu} \cdot e\left(g^{v_\tau}, \left(V_0 \prod_{j=1}^{k} V_j^{\tau_j}\right)^{s\mu}\right) \cdot e(g,g)^{b^2 t\mu s}}{e\left(g^{\alpha}, g^{s\mu}\right) \cdot e\left(g^{b^2 t}, g^{s\mu}\right) \cdot e\left(\left(V_0 \prod_{j=1}^{k} V_j^{\tau_j}\right)^{v_\tau}, g^{s\mu}\right)} \\ &= \mathscr{M}.\end{aligned} \tag{9}$$

(v) $\mathrm{CTUpdate}(\mathrm{PK}, \mathrm{CT}, T_c, \mathscr{R}', \mathbb{A} = (M, \rho))$: the ciphertext update algorithm takes as input the ciphertext CT, the decryptable time period T_c, and a new revocation list $\mathscr{R}'$ such that $\mathscr{R} \subseteq \mathscr{R}'$. Denote $\mathscr{R}' = (\mathrm{ID}_1, \ldots, \mathrm{ID}_r, \ldots, \mathrm{ID}_{r+r'})$ with $r + r'$ revoked users. It takes an LSSS access structure $\mathbb{A} = (M, \rho)$, where M is an $l \times n$ matrix and ρ is a function maps rows of M into attributes. The algorithm chooses a random vector $v' = (s', y_2', \ldots, y_n') \in \mathbb{Z}_p^n$ to share the encryption exponent s'. For $i = 1, \ldots, l$, it calculates $\lambda_i' = v' \cdot M_i$, where the vector M_i corresponds to the i-th row of M. Let ID_j denote the j-th identity. It also chooses random c such that $\mu' = \mu_1', \ldots, \mu_r', \ldots, \mu_{r+r'}' \in \mathbb{Z}_p$ and computes

$$\begin{aligned}\widetilde{C}_0 &= C_0 \cdot e(g,g)^{\alpha s'\mu'} = \mathscr{M} e(g,g)^{\alpha\left(s\mu + s'\mu'\right)}, \\ \widetilde{C}_0' &= C_0' \cdot g^{s'\mu'} = g^{\left(s\mu + s'\mu'\right)}, \\ \widetilde{C}_0'' &= C_0'' \cdot \left(V_0 \prod_{j=1}^{k} V_j^{\tau_j}\right)^{s'\mu'} = \left(V_0 \prod_{j=1}^{k} V_j^{\tau_j}\right)^{\left(s\mu + s'\mu'\right)}, \\ \widetilde{C}_{i,j} &= C_{i,j} \cdot g^{b\lambda_i'\mu_j'} = g^{b\left(\lambda_i\mu_j + \lambda_i'\mu_j'\right)}, \\ \widetilde{C}_{i,j}' &= C_{i,j}' \cdot \left(g^{b^2 \mathrm{ID}_j} h_{\rho(i)}^{b}\right)^{\lambda_i'\mu_j'} = \left(g^{b^2 \mathrm{ID}_j} h_{\rho(i)}^{b}\right)^{\left(\lambda_i\mu_j + \lambda_i'\mu_j'\right)}.\end{aligned} \tag{10}$$

Then, $\mathrm{CT}' = \left\{\widetilde{C}_0, \widetilde{C}_0', \widetilde{C_0''}, \widetilde{C}_{i,j}, \widetilde{C}_{i,j}'\right\}$ along with a description of the revoked set $\mathscr{R}$, the access structure $\mathbb{A} = (M, \rho)$, and the time periods T_c.

5. Security Analysis

Our construction security is based on the modified decisional q-parallel-BDHE assumption. It is apparent that the updated ciphertext has the same distribution as the original ciphertext, so we only prove the security associated with the original ciphertext.

Theorem1. *Suppose the modified decisional q-parallel-BDHE assumption holds. Then, no PPT adversary can selectively break our system in with a challenge matrix of size $l^* \times n^*$, where $l^*, n^* < q$, a challenge revocation list $\mathscr{R}^*$ where $\mathscr{R}^* < q - 2$ and a challenge time T_c^* with z-ary representation $\tau^* = (\tau_1^*, \ldots, \tau_{k^*}^*)$ for some $k^* < \mathtt{T}$ such that $\mathtt{T} < q$.*

Proof. Suppose there is an adversary $\mathscr{A}$ with nonnegligible advantage $\varepsilon = \mathrm{Adv}_{\mathscr{A}}$ against our scheme in the selective security game. Then, simulator $\mathscr{B}$ can solve the modified decisional q-parallel-BDHE problem with nonnegligible advantage.

(i) Init: the simulator $\mathscr{B}$ takes in a modified decisional q-parallel-BDHE problem challenge $\{\mathbf{y}, T\}$:

$$\begin{aligned}\mathbf{y} = \Big\{&g, g^{s}, g^{a}, \ldots, g^{(a^q)}, g^{(a^{q+2})}, \ldots, g^{(a^{2q})}, \\ &\forall_{1\le j\le q} g^{a/b_j}, \ldots, g^{a^q/b_j}, g^{a^{q+2}/b_j}, \ldots, g^{a^{2q}/b_j}, \\ &\forall_{1\le j\le q} g^{a\cdot s/b_j}, \ldots, g^{a^q\cdot s/b_j}\Big\},\end{aligned} \tag{11}$$

and decides if $T = T = e(g,g)^{s\cdot a^{q+1}}$ using the adversary $\mathscr{A}$. Then, the adversary $\mathscr{A}$ declares the challenge time T_c^* with z-ary representation $\tau^* = (\tau_1^*, \ldots, \tau_{k^*}^*)$ for some $k^* \le \mathrm{T}$ and the challenge revocation list $\mathscr{R}^*$, where $|\mathscr{R}^*| \le q - 2$. $\mathscr{A}$ also gives the challenge access structure (M^*, ρ^*) to the simulator $\mathscr{B}$, where M^* is $l^* \times n^*$. Let $M^* = (M_1^*, M_2^*, \ldots, M_{l^*}^*)^T$, where each row vector $M_i^* = (M_{i,1}^*, M_{i,2}^*, \ldots, M_{i,n^*}^*)$ for $1 \le i \le l^*$.

(ii) Setup: the simulator $\mathscr{B}$ chooses a random value α' and lets $e(g,g)^{\alpha} = e(g,g)^{\alpha'} e(g^{a}, g^{a^q})$ to implicitly set $\alpha = \alpha' + a^{q+1}$. Moreover, it also implicitly sets $b = a$ by computing the public parameters as $g^{b} = g^{a}, g^{b^2} = g^{a^2}$. To embed the revocation identification ID_c and the challenge access structure into the public parameters $h_1, h_2, \ldots, h_U$, we let the challenge matrix $\mathbf{M}^*$ as a row vector set and divide it into three subsets M', M'', M''' such that $M' \bigcup M'' \bigcup M''' = M^*$ and $M' \cap M'' \cap M''' = \varnothing$. Specially, M', M'', and M''' are initially set to be empty set. Define the n-dimension vectors $\mathbf{e} = (1, 0, \ldots, 0)$ and $\mathbf{u} = (a^2, a^3, \ldots, a^{n+1})$. For $i = 1$ to l^*, if $\mathbf{M}_i^*$ is linearly independent on M' and $\mathbf{e}$ cannot be linearly expressed by $M' \bigcup \mathbf{M}_i^*$, then we merge $\mathbf{M}_i^*$ into M'; if $\mathbf{M}_i^*$ is linearly independent on M' and $\mathbf{e}$ can be linearly expressed by $M' \bigcup \mathbf{M}_i^*$, then we merge $\mathbf{M}_i^*$ into M'''; if $\mathbf{M}_i^*$ is dependent on M', then we merge $\mathbf{M}_i^*$ into M''. As a result, M' is a linear independent vector group, while each vector in M'' can be linearly expressed by M'. Although $\mathbf{e}$ cannot be spanned by M', it can be linearly expressed by M' merged with each vector in M'''. Therefore, each vector in M^* can be linearly expressed by $M' \bigcup \mathbf{e}$.

Next, we describe how the simulator $\mathscr{B}$ programs the public parameters $h_1^b, h_2^b, \ldots, h_U^b$. Let X denote the set of indices i, such that $\rho(i) = x$. Assume that there are m vectors in M' and let $M' = (\mathbf{M}_1', \mathbf{M}_2', \ldots, \mathbf{M}_m')^T$. For each $i \in X$, its corresponding row vector $\mathbf{M}_i^*$ can be written as $\varepsilon_{i,0}\mathbf{e} + \varepsilon_{i,1}\mathbf{M}_1' + \cdots + \varepsilon_{i,m}\mathbf{M}_m'$, where $(\varepsilon_{i,0}, \varepsilon_{i,1}, \ldots, \varepsilon_{i,m}) \in \mathbb{Z}_p^{m+1}$. For each $\mathbf{M}_i^*$, we define a corresponding vector $\mathbf{M}_i$, where $\mathbf{M}_i = \varepsilon_{i,1}\mathbf{M}_1' + \cdots + \varepsilon_{i,m}\mathbf{M}_m'$. As a result, we get a new vector group $M' = (\mathbf{M}_1, \mathbf{M}_2, \ldots, \mathbf{M}_{l^*})$, and each $\mathbf{M}_i$ is in the span of M'. By choosing a random value z_x, the simulator $\mathscr{B}$ programs h_x and h_x^b as

$$\begin{aligned} h_x &= g^{z_x} \cdot g^{-a\sum_{l=1}^{r} \mathrm{ID}_{c_l}} \cdot \prod_{i \in X} g^{(\varepsilon_{i,0}\mathbf{e}+\varepsilon_{i,1}\mathbf{M}_1'+\cdots+\varepsilon_{i,m}\mathbf{M}_m')\cdot\overrightarrow{u} \longrightarrow /b_i} \\ &= g^{z_x} \cdot g^{-a\sum_{l=1}^{r} \mathrm{ID}_{c_l}} \cdot \left(\prod_{i \in X} \prod_{j=1}^{n} g^{M_{i,j} a^{j+1/b_i}} \right), \\ h_x^b &= g^{z_x} \cdot g^{-a^2\sum_{l=1}^{r} \mathrm{ID}_{c_l}} \cdot \left(\prod_{i \in X} \prod_{j=1}^{n} g^{M_{i,j} a^{j+1/b_i}} \right). \end{aligned} \tag{12}$$

If X is an empty set, it sets $h_x^b = g^{z_x}$. And the simulator $\mathscr{B}$ also randomly chooses $\xi_0, \xi_1, \ldots, \xi_T \in \mathbb{Z}_p$ and defines $V_j = g^{\xi_j a^{q-j+1}}$ and $V_0 = V_o = \prod_{j=1}^{k^*} V_j^{-\tau_j^*} g^{\xi_0}$ for $j \in [1, \mathrm{T}]$. Then, $\mathscr{B}$ publishes the above parameters $(g, g^b, g^{b^2}, h_1^b, \ldots, h_U^b, V_0, \ldots, V_{\mathrm{T}}, e(g,g)^{\alpha})$ as the public key and sends it to $\mathscr{A}$. We observe that the public parameters are distributed randomly as the real system and both the revoked identification and the challenge matrix are reflected in the simulation's contribution of the parameter h_x^b.

(i) Phase1: adversary $\mathscr{A}$ makes repeated private keys queries corresponding to the tuple of identity, attributes, and time (ID, S, T) such that at least one of the following requirements is satisfied:

  (i) The attributes set S^* satisfies the access structure $\mathbb{A}^*$ and the corresponding identity $\mathrm{ID} \in \mathscr{R}^*$

  (ii) T_c^* and all its prefixes are not in $\mathbb{T}$, the set-cover of T

  We separate into two cases:

(i) Case 1: the attributes set S^* satisfies the access structure $\mathbb{A}^*$ and the corresponding identity $\mathrm{ID} \in \mathscr{R}^*$. Since each $\mathbf{M_i}$ is in the span of M_1^* and $\mathbf{e}$ is not in the span of M_1^*, we can still find a vector ω with $\omega_1 = -1$ and $\omega \cdot \mathbf{M_i} = 0$, where $1 \le i \le l^*$. The simulator $\mathscr{B}$ chooses a random value r' and computes the private key as

$$L = g^{r' + \omega \cdot \mathbf{v}} = g^{r'} \prod_{i=1}^{l^*} \left(g^{a^{q-i}} \right)^{\omega_i}, \tag{13}$$

which implicitly sets the random t as $t = r' + \omega \cdot \mathbf{v} = r' + \omega_1 a^{q-1} + \omega_1 a^{q-2} + \cdots + \omega_n a^{q-n^*}$, where $\mathbf{v} = (a^{q-1}, a^{q-2}, \ldots, a^{q-n^*+2})$. So, it can cancel out the unknown term of the form g^{q+1} in g^{α} when creating the K component in the private key as

$$K = g^{\alpha'} g^{a^2 r'} \prod_{i=0}^{n-2} \left(g^{a^{q+i}} \right)^{\omega_i}. \tag{14}$$

Next, it performs this by setting

$$D_0 = g^{r'} \prod_{i=1}^{n} \left(g^{a^{q+1-i}} \right)^{\omega_i} = g^{t}. \tag{15}$$

In order to prevent the appearance of the term of the form $g^{a^{q+1}}$, it sets the private component K_x as

$$\begin{aligned} K_x &= \left(g^{z_x} g^{a\left(\mathrm{ID} - \sum_{m=1}^{r} \mathrm{ID}_{c_m}\right)} \cdot \prod_{i \in X} g^{\mathbf{M_i}\cdot\overrightarrow{u}/b_i} \right)^{r+\omega\cdot\mathbf{v}} \\ &= g^{z_x (r+\omega\cdot\mathbf{v})} g^{a\left(\mathrm{ID} - \sum_{m=1}^{r} \mathrm{ID}_{c_m}\right)(r+\omega\cdot\mathbf{v})} \\ &\quad \cdot \prod_{i \in X} g^{\left(\varepsilon_{i,0} e + \varepsilon_{i,1}\mathbf{M}_1' + \cdots + \varepsilon_{i,m}\mathbf{M_m}'\right)(r+\omega\cdot\mathbf{v})/b_i} \\ &= g^{z_x (r+\omega\cdot\mathbf{v})} g^{a\left(\mathrm{ID} - \sum_{m=1}^{r} \mathrm{ID}_{c_m}\right)(r+\omega\cdot\mathbf{v})} \\ &\quad \cdot \prod_{i \in X} g^{\left(M_{i,1}a^2 + \cdots + M_{i,n}a^n\right) r/b_i} \\ &\quad \cdot \sum_{i \in X} \prod_{j=1}^{n} \prod_{k=1, k \neq j}^{n} g^{M_{i,j}\omega_k a^{q+j-k+1}/b_i}, \end{aligned} \tag{16}$$

$\mathscr{B}$ randomly chooses $v_\tau \in \mathbb{Z}_p$ and sets $D_{0,\tau} = g^{v_\tau}$ for all $\tau = (\tau_1, \ldots, \tau_{k_\tau}) \in \mathbb{T}$. Then, it computes

$$\begin{aligned} D_{1,\tau} &= g^{\alpha'} g^{ar'} \prod_{i=1}^{n^*} \left(g^{a^{q+2-i}} \right)^{\omega_i} \left(V_0 \prod_{j=1}^{k_\tau} V_j^{\tau_j} \right)^{v_\tau} \\ &= g^{\alpha'} g^{a^{q+1}} g^{ar'} g^{-a^{q+1}} \prod_{i=2}^{n^*} \left(g^{a^{q+2-i}} \right)^{\omega_i} \left(V_0 \prod_{j=1}^{k_\tau} V_j^{\tau_j} \right)^{v_\tau} \\ &= g^{\alpha} g^{ar'} g^{\omega_1 \cdot a^{q+1}} \prod_{i=2}^{n^*} \left(g^{a^{q+2-i}} \right)^{\omega_i} \left(V_0 \prod_{j=1}^{k_\tau} V_j^{\tau_j} \right)^{v_\tau} \\ &= g^{\alpha} \left(g^{r'} \prod_{i=1}^{n^*} \left(g^{a^{q+1-i}} \right)^{\omega_i} \right)^{a} \left(V_0 \prod_{j=1}^{k_\tau} V_j^{\tau_j} \right)^{v_\tau} \\ &= g^{\alpha} g^{at} \left(V_0 \prod_{j=1}^{k_\tau} V_j^{\tau_j} \right)^{v_\tau} \\ &= g^{\alpha} g^{bt} \left(V_0 \prod_{j=1}^{k_\tau} V_j^{\tau_j} \right)^{v_\tau}, \end{aligned} \tag{17}$$

$\mathscr{B}$ also computes $\left\{ L_{j,\tau} = V_j^{v_\tau} \right\}_{j=k_\tau+1,\ldots,\mathrm{T},\tau\in\mathbb{T}}$.

(ii) Case 2: T_c^* and all its prefixes are not in $\mathbb{T}$, the set-cover of T. For all τ=($\tau = (\tau_1, \ldots, \tau_{k_\tau}) \in \mathbb{T}$), first define $\tau_{k_{\tau+1}} = \cdots = \tau_q = 0$ and $\tau_{k^*+1}^* = \cdots = \tau_q^* = 0$. There exists a smallest index $k' \le k^*$ such that $\tau_{k'} \neq \tau_{k'}^*$. Simulator $\mathscr{B}$ randomly selects $v_\tau \in \mathbb{Z}_p$ and implicitly defines $\widetilde{v_\tau} = (a^{k'}/\xi_{k'}(\tau_{k'}^* - \tau_{k'})) + v_\tau$. It performs this by setting

$$D_{0,\tau} = g^{\left(a^{k'}/\xi_{k'}\left(\tau^*_{k'} - \tau_{k'}\right)\right) + \nu_\tau}. \tag{18}$$

$\mathscr{B}$ then chooses a random element $t \in \mathbb{Z}_p$ and sets $D_{0,\tau} = g^t$. For all τ, it computes

$$\begin{aligned}
D_{1,\tau} &= g^{\alpha' + \alpha_0 t + \nu_\tau \xi_0}\left(g^{a^{q-k'+1}}\right)^{\nu_\tau \xi_{k'}\left(\tau_{k'} - \tau^*_{k'}\right)} \cdot \left(g^{a^{k'}}\right)^{\left(\xi_0/\xi_{k'}\left(\tau^*_{k'} - \tau_{k'}\right)\right)} \\
&\quad \cdot \prod_{j=1}^{k_\tau - k' + 1}\left(g^{a^{q-j+1}}\right)^{\left(\xi_j + k'\tau^*_j + k'/\xi_{k'}\left(\tau^*_{k'} - \tau_{k'}\right)\right)} \cdot \prod_{j=k'+1}^{k_\tau+1}\left(g^{a^{q-j+1}}\right)^{\xi_j \tau^*_j \nu_\tau} \\
&= g^{\alpha'} g^{\alpha_0 t} g^{\nu_\tau\left(\xi_0 + \xi_{k'} a^{q-k'+1}\left(\tau_{k'} - \tau^*_{k'}\right)\right)} \cdot g^{\left(\xi_0 a^{k'}/\xi_{k'}\left(\tau^*_{k'} - \tau_{k'}\right)\right)} \\
&\quad \cdot \prod_{j=k'+1}^{k_\tau+1} g^{\left(\xi_j a^{q-j+1+k'}\tau^*_j/\xi_{k'}\left(\tau^*_{k'} - \tau_{k'}\right)\right)} \cdot \prod_{j=k'+1}^{k_\tau+1} g^{\xi_j a^{q-j+1}\tau^*_j \nu_\tau} \\
&= g^{\alpha' + a^{q+1}} g^{\alpha_0 t} \cdot g^{\left(\xi_0 + \xi_{k'} a^{q-k'+1}\left(\tau_{k'} - \tau^*_{k'}\right)\right)\left(\left(a^{k'}/\xi_{k'}\left(\tau^*_{k'} - \tau_{k'}\right)\right) + \nu_\tau\right)} \\
&\quad \cdot \prod_{j=k'+1}^{k_\tau+1} g^{\xi_j a^{q-j+1}\tau^*_j\left(\left(a^{k'}/\xi_{k'}\left(\tau^*_{k'} - \tau_{k'}\right)\right) + \nu_\tau\right)} \\
&= g^{\alpha} g^{\alpha_0 t} g^{\left(\xi_0 + \xi_{k'} a^{q-k'+1}\left(\tau_{k'} - \tau^*_{k'}\right)\right)\widetilde{\nu_\tau}} \cdot \prod_{j=k'+1}^{k_\tau+1} g^{\xi_j a^{q-j+1}\tau^*_j \widetilde{\nu_\tau}} \\
&= g^{\alpha} g^{\alpha_0 t}\left(V_0 \prod_{j=1}^{k'} V_j^{\tau_j}\right)^{\widetilde{\nu_\tau}} \prod_{j=k'+1}^{k_\tau+1} V_j^{\tau_j \widetilde{\nu_\tau}} \\
&= g^{\alpha} g^{\alpha_0 t}\left(V_0 \prod_{j=1}^{k_\tau+1} V_j^{\tau_j}\right)^{\widetilde{\nu_\tau}} \\
&= g^{\alpha} g^{\alpha_0 t}\left(V_0 \prod_{j=1}^{k_\tau} V_j^{\tau_j}\right)^{\widetilde{\nu_\tau}} \left(\because \tau_{k_\tau+1=0}\right).
\end{aligned} \tag{19}$$

Simulator $\mathscr{B}$ also computes $\{K_x\}$ and

$$\begin{aligned}
L_{j,\tau} &= g^{\left(\xi_j a^{q+1+k'-j}/\xi_{k'}\left(\tau^*_{k'} - \tau_{k'}\right)\right) + \nu_\tau \xi_j a^{q-j+1}} \\
&= \left(g^{\xi_j a^{q-j+1}}\right)^{\left(a^{k'}/\xi_{k'}\left(\tau^*_{k'} - \tau'_k\right)\right) + \nu_\tau} \\
&= V_j^{\widetilde{\nu_\tau}}.
\end{aligned} \tag{20}$$

(iii) Challenge: adversary $\mathscr{A}$ submits two equal length messages $\mathscr{M}_0$ and $\mathscr{M}_1$ with the matrix M^* of dimension at most n columns to $\mathscr{B}$. $\mathscr{B}$ flips a random coin b and encrypts $\mathscr{M}_b$ under the access structure $\mathbb{A}^*$, the revocation list $\mathscr{R}^*$, and the time T^*_c with z-ary representation τ^*. It chooses random values $\mu_1, \mu_2, \ldots, \mu_r$ such that $\mu = \mu_1 + \mu_2 + \cdots + \mu_r$ and creates the ciphertext components

$$\begin{aligned}
C_0 &= \mathscr{M}_b\left(T \cdot e\left(g^s, g^{\alpha'}\right)\right)^{\mu}, \\
C'_0 &= g^{s\mu}.
\end{aligned} \tag{21}$$

For C''_0, observe that since the challenge time is $(\tau^*_1, \ldots, \tau^*_{k^*})$, the g^{a^i} terms in V_i are cancelled out. Then, it sets $C''_0 = (g^s)^{\xi_0}$. $\mathscr{B}$ also chooses random value $y'_2, y'_3, \ldots, y_{n^*}' \in \mathbb{Z}_p$ and shares the secret s using the vector $\mathbf{x} = (s, y'_2, y'_3, \ldots, y_{n^*}') \in \mathbb{Z}^n_p$. Next, it calculates

$$\begin{aligned}
\lambda_k &= \mathbf{x} \cdot \left(\varepsilon_{i,0}\mathbf{e} + \varepsilon_{i,1}\mathbf{M}^*_1 + \cdots + \varepsilon_{i,m}\mathbf{M}^*_{\mathbf{m}}\right) \\
&= \mathbf{x} \cdot \left(\varepsilon_{i,0}\mathbf{e} + \mathbf{M}^*_{\mathbf{k}}\right).
\end{aligned} \tag{22}$$

And it generates the ciphertext component $C_{i.j}$ as

$$C_{i.j} = g^{asv_r\left(M_{k,1} + \varepsilon_{k,0}\right)} \cdot \prod_{i=2}^{n} g^{M_{k,i} y'_i}. \tag{23}$$

Table 1: Comparison of features.

Scheme	Instant revoke	Short revocation list	Key update	Ciphertext update
[23]	✓	×	×	×
[29]	✓	✓	✓	×
Ours	✓	✓	✓	✓

Table 2: Efficiency comparison.

Scheme	PK size	SK size	Ciphertext size	Decryption time (of pairing)
[23]	$(\|U\| + 3)\mathbb{G}_1 + \mathbb{G}_1$	$(\|S\| + 2)\mathbb{G}_1$	$(2lR + 1)\mathbb{G}_1 + \mathbb{G}_2$	$2\|I\|R + 1$
[29]	$(\|U\| + R + \mathtt{T} + 3)\mathbb{G}_1 + \mathbb{G}_2$	$(\|S\| + Z + R + 1)\mathbb{G}_1$	$(l + 3)\mathbb{G}_1 + \mathbb{G}_2$	$2\|I\| + 4$
Our	$(\|U\| + \mathtt{T} + 3)\mathbb{G}_1 + \mathbb{G}_2$	$(\|S\| + Z + 1)\mathbb{G}_1$	$(2lR' + 1)\mathbb{G}_1 + \mathbb{G}_2$	$2\|I\|R' + 2$

U: the max number of attributes in the system, $\mathbb{G}_1$: number of $\mathbb{G}_1$ elements, $\mathbb{G}_2$: number of $\mathbb{G}_2$ elements, S: the set of attributes created for a specific user, l: the number of attributes involved in the encryption process, I: the identity set defined in the system, R: the max number of the revoked users, R': the length of the revocation list, $\mathtt{T}$: the depth of the time tree, Z: best case $Z = 2$, and worst case $Z = (\mathtt{T}(\mathtt{T} + 2)/2)$.

(i) For $k = 1, 2, \ldots, n$, it defines X_k as the set of the index i such that $\rho(i) = \rho(k)$. Finally, $\mathscr{B}$ builds the ciphertext component $C^*_{i,j}$ as

$$\begin{aligned} C^*_{i,j} &= \left(g^{a^2 \sum_{m=1}^{r} \mathrm{ID}_{c_m}} g^{z_x} g^{-a^2 \sum_{l=1}^{r} \mathrm{ID}_{c_l}} \cdot \prod_{i \in X_k} g^{\mathbf{M_i} \cdot \mu / b_i} \right)^{\lambda_k \nu_\gamma} \\ &= g^{z_x \lambda_k \nu_\gamma} \prod_{i \in X_k} g^{\left(M_{i,1} a^2 + M_{i,2} a^3 + \cdots + M_{i,n} a^{n+1}\right)^{\lambda_k \nu_\gamma / b_i}}. \end{aligned} \tag{24}$$

(ii) Phase 2: this phase is completely the same as the Phase 1.

(iii) Guess: the adversary $\mathscr{A}$ will finally output a guess b' of b. $\mathscr{B}$ outputs 0 to guess $T = e(g, g)^{a^{q+1} s}$ if $b' = b$; otherwise, it outputs 1. When T is a tuple, $\mathscr{B}$ gives a perfect simulation, so we have that the advantage of the simulator $\mathscr{B}$ is the same as the advantage of the adversary $\mathscr{A}$. Therefore, we have

$$\Pr\left[\mathscr{B}\left(\mathbf{y}, T = e(g, g)^{a^{q+1} s}\right) = 0\right] = \frac{1}{2} + \mathrm{Adv}_{\mathscr{A}}. \tag{25}$$

The message $\mathscr{M}_b$ is completely hidden from the adversary when T is a random group element, so we have $\Pr[\mathscr{B}(\mathbf{y}, T = R) = 0] = (1/2)$. Therefore, if $\mathscr{A}$ could attack scheme with nonnegligible advantage, then $\mathscr{B}$ can also play the modified decisional q-parallel-BDHE game with nonnegligible advantage. □

6. Performance Analysis

In this section, we first give a functional comparsion between our scheme and other schemes [23, 29] in Table 1.

Our scheme can implement user direct revocation, maintain a short revocation list, and update ciphertext. Compared with [23], our scheme can maintain a short revocation list and update ciphertext. Compared with [29], our scheme can update ciphertext. The ciphertext update can provide the encrypted data confidentiality by disqualifying the revoked users' access to the encrypted data, especially that generated previously. We can periodically run a ciphertext update algorithm and do not need to execute a key update algorithm frequently because users have a reasonable validity time.

Next, we mainly analyze the efficiency of the proposed scheme compared with [23, 29] in Table 2.

As shown in Table 2, the efficiency of the proposed scheme is a little lower than scheme [23], but we can reduce the size of ciphertext by maintaining a short revocation list. In addition, the efficiency of our scheme is lower than scheme [29] in terms of the ciphertext size and the decryption time of pairing, but our scheme is more efficient in the size of *PK* and *SK*. The number of exponentiation operations in the KenGen algorithm in the scheme [29] is R times more than our scheme, and the number of exponentiation operations in the Encrypt algorithm in the scheme [29] is r times more than our scheme. Our scheme is practical that it can revoke users immediately, maintain a short revocation list, and update ciphertext, but loses the advantage of efficiency in the ciphertext size and the decryption time.

7. Conclusion

In this work, we propose a user R-CP-ABE scheme with ciphertext update. The scheme can implement user direct revocation, maintain a short revocation list, and update ciphertext by incorporating the identity-based and time-based revocable technique. We provide a ciphertext update mechanism, using only publicly available information, to disqualify the revoked users from accessing previously encrypted data. Our scheme supports the key update function for the nonrevoked users when their validity time expires. Once the validity time expires, the user's key becomes invalid and cannot decrypt any newly generated ciphertext after the expiry date. The security is based on the modified decisional q-parallel bilinear Diffie–Hellman Exponent problem. In the security model, we consider a strong adversary that can query the secret key of a user whose attribute set satisfies the challenge ciphertext access policy and whose identity is in the revocation list. In the future research, we will consider a more efficient mechanism for the user revocation and ciphertext update.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported in part by the National Key R&D Program of China (no. 2017YFB0802000), the National Nature Science Foundation of China (nos. U1705264, 61972124, 61672030, 11974096, and 61702067), the Zhejiang Provincial Natural Science Foundation of China (no. LY19F020019), and the Research Foundation of Hangzhou Normal University (no. 2017QDL002).

References

[1] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in *Proceedings of the 13th ACM Conference on Computer and Communications Security*, pp. 89–98, Chicago, IL, USA, 2006.

[2] R. Ostrovsky, A. Sahai, and B. Waters, "Attribute-based encryption with non-monotonoc access structures," in *Proceedings of the 14th ACM Conference on Computer and Communications Security 2007*, pp. 195–203, Alexandria, VA, USA, 2007.

[3] A. Sahai and B. Waters, "Fuzzy identity-based encryption," in *EUROCRYPT 2005*, R. Cramer, Ed., vol. 196, pp. 457–473, Springer, Berlin, Germany, 2005.

[4] N. Attrapadung, B. Libert, and E. de Panafieu, "Expressive key-policy attribute-based encryption with constant-size ciphertexts," in *Proceedings of the Public Key Cryptography–PKC 2011*, Springer, Berlin, Germany, 2011.

[5] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in *Proceedings of the IEEE Symposium on IEEE*, pp. 321–334, Washington, DC, USA, 2007.

[6] L. Cheung and C. Newport, "Provably secure ciphertext policy ABE," in *Proceedings of the ACM Conference on Computer and Communications Security*, pp. 456–465, CCS), Chicago, IL, USA, 2007.

[7] B. Waters, *Ciphertext-Policy Attribute-Based Encryption: An Expressive, Efficient, And Provably Secure Realization*, pp. 53–70, University of Texas Springer, Austin, TX, USA, 2011.

[8] J. Hong-Yong, C. Yue, M. Xiu-Qing et al., "Ciphertext-policy attribute-based encryption with non-monotonic access structure," in *Proceedings of the International Colloquium on Computing, Communication, Control, and Management*, Guangzhou, China, 2008.

[9] K. Liang, M. H. Au, J. K. Liu et al., "A dfa-based functional proxy re-encryption scheme for secure public cloud data sharing," *IEEE Transactions on Information Forensics and Security*, vol. 9, no. 10, pp. 1667–1680, 2014.

[10] K. Liang, M. H. Au, J. K. Liu et al., "A secure and efficient ciphertext-policy attribute-based proxy re-encryption for cloud data sharing," *Future Generation Computer Systems*, vol. 52, pp. 95–108, 2015.

[11] S. Wang, K. Liang, J. K. Liu, J. Chen, J. Yu, and W. Xie, "Attribute-based data sharing scheme revisited in cloud computing," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 8, pp. 1661–1673, 2016.

[12] C. Zuo, J. Shao, J. K. Liu, G. Wei, and Y. Ling, "Fine-grained two-factor protection mechanism for data sharing in cloud storage," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 1, pp. 186–196, 2018.

[13] Q. Huiling, L. Jiguo, Z. Yichen, and H. Jinguan, "Privacy-preserving personal health record using multi-authority attribute-based encryption with revocation," *International Journal of Information Security*, vol. 14, no. 6, pp. 487–497, 2015.

[14] M. H. Au, T. H. Yuen, J. K. Liu et al., "A general framework for secure sharing of personal health records in cloud system," *Journal of Computer and System Sciences*, vol. 90, pp. 46–62, 2017.

[15] K. He, J. Weng, J. K. Liu, W. Zhou, and J. Liu, "Efficient fine-grained access control for secure personal health records in cloud computing," in *Proceedings of the International Conference on Network and System Security*, pp. 65–79, Springer, Taipei, Taiwan, 2016.

[16] J. Liu, X. Huang, and J. K. Liu, "Secure sharing of personal health records in cloud computing: ciphertext-policy attribute-based signcryption," *Future Generation Computer Systems*, vol. 52, pp. 67–76, 2015.

[17] L Jiguo, "Flexible and fine-grained attribute-based data storage in cloud computing," *Services Computing IEEE Transactions on*, vol. 10, no. 5, pp. 785–796, 2017.

[18] F. Xhafa, J. Wang, X. Chen, J. K. Liu, J. Li, and P. Krause, "An efficient PHR service system supporting fuzzy keyword search and fine-grained access control," *Soft Computing*, vol. 18, no. 9, pp. 1795–1802, 2014.

[19] A. Boldyreva, V. Goyal, and V. Kumar, "Identity-based encryption with efficient revocation," in *Proceedings of the ACM Conference on Computer and Communications Security*, pp. 417–426, ACM, Chicago, IL, USA, 2008.

[20] A. Sahai, H. Seyaloiglu, and B. Waters, "Dynamic credentials and ciphertext delegation for attribute-based encryption," in *Proceedings of the IEEE Symposium on Security and Privacy*, pp. 199–217, Springer, 2012.

[21] N. Attrapadung and H. Imai, "Conjunctive broadcast and attribute-based encryption," in *Proceedings of the International Conference on Cryptography and Coding*, vol. 5671, pp. 248–265, Springer, Berlin, Germany, 2009.

[22] Z. Liu and D. S. Wong, "Practical ciphertext-policy attribute-based encryption: traitor tracing, revocation and large universe," *The Computer Journal*, vol. 59, no. 7, pp. 127–146, 2015.

[23] W. Weijia, W. Zhijie, L. Bing, D. Qiuxiang, and H. Dijiang, "IR-CP-ABE: identity revocable ciphertext-policy attribute-based encryption for flexible secure group-based communication," *IACR Cryptology ePrint Archive*, vol. 1100, p. 2017, 2017.

[24] J. Ye, W. Zhang, S. Wu, Y. Gao, and J. Qiu, "Attribute-based fine-grained access control with user revocation," in *Proceedings of the EurAsia*, pp. 586–595, Springer, Bali, Indonesia, 2014.

[25] S. Yu, C. Wang, K. Ren, and W. Lou, "Attribute based data sharing with attribute revocation," in *Proceedings of the ASIACCS*, pp. 261–270, ACM, Beijing, China, 2010.

[26] A. Balu and K. Kuppusamy, "Ciphertext-policy attribute-based encryption with user revocation support," in *Proceedings of the QShine, Volume 115 of Lecture Notes of the Institute for Computer Sciences, Social Informatices and Telecommunications Engineering*, pp. 696–705, Springer, Berlin, Germany, 2013.

[27] X. Liang, R. Lu, X. Lin, and X. Shen, "Ciphertext policy attribute based encryption with efficient revocation," Technical Report, University of Waterloo, Waterloo, Canada, 2011.

[28] X. Xie, H. Ma, J. Li, and X. Chen, "New ciphertext-policy attribute-based access control with efficient revocation," in *Proceedings of the Information and Communication Technology - EurAsia Conference*, pp. 373–382, Springer, Bali, Indonesia, 2013.

[29] J. K. Liu, T. H. Yuen, P. Zhang, and K. Liang, "Time-based direct revocable ciphertext-policy attribute-based encryption with short revocation list," in *Applied Cryptography and Network Security*Springer, Cham, Switzerland, 2018.

[30] A. Beimel, "Schemes for secret sharing and key distribution," in *Proceedings of the Technion-Israel Institute of technology, Faculty of computer science*, Haifa, Israel, 1996.

[31] A. Lewko, A. Sahai, and B. Waters, "Security and privacy (SP)," in *Proceedings of the 2010 IEEE Symposium on IEEE*, pp. 273–285, San Diego, CA, USA, 2010.