IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received April 15, 2019, accepted May 2, 2019, date of publication May 22, 2019, date of current version June 4, 2019.

*Digital Object Identifier 10.1109/ACCESS.2019.2918434*

# Updatable Ciphertext-Policy Attribute-Based Encryption Scheme With Traceability and Revocability

**ZHENHUA LIU[1], JING XU[1], YAN LIU[1], AND BAOCANG WANG[2]**
[1]School of Mathematics and Statistics, Xidian University, Xi'an 710071, China
[2]School of Information Engineering, Xuchang University, Xuchang 461000, China

Corresponding author: Zhenhua Liu (zhualiu@hotmail.com)

This work was supported in part by the National Key R&D Program of China under Grant 2017YFB0802000, in part by the National Natural Science Foundation of China under Grant 61807026, in part by the Natural Science Basic Research Plan in Shaanxi Province of China under Grant 2019JM-198, in part by the Plan For Scientific Innovation Talent of Henan Province under Grant 184100510012, and in part by the Program for Science and Technology Innovation Talents in the Universities of Henan Province under Grant 18HASTIT022.

**ABSTRACT** Ciphertext-policy attribute-based encryption (CP-ABE) can offer fine-grained access control over encrypted data, which is suitable for complex commercial applications. However, since the same decryption privileges could be shared by multiple users in the one-to-many encryption mechanism, it is dangerous that a malicious user misuses his secret key but cannot be traced. In addition to further security, when the malicious user has been caught, it is required to revoke him from the system. To address these problems, we propose a novel updatable CP-ABE scheme supporting white-box traceability and traitor revocation. In the proposed scheme, a ''fixed point'' is embedded into the user's secret key to achieving the traceability and each user is assigned with a unique identifier for revocation. Moreover, the secret exponent used to encrypt a message is divided into two parts: one is assigned to access policy and the other to the revocation list. Therefore, only a part of the ciphertext components needs to be updated when the revocation list is changed, which greatly simplifies the process of ciphertext update. Compared to the previous works, our scheme is more efficient, and can achieve valid revocation and ciphertext update. In addition, the traceability of the proposed scheme is depended on the $l$-Strong Difffie–Hellman assumption, and the indistinguishability security under selective access policy and chosen-plaintext attacks in the standard model is reduced to the Decisional $q$-Bilinear Diffie–Hellman assumption. Furthermore, the experimental results show that the proposed scheme is efficient.

**INDEX TERMS** Cloud storage, access control, attribute-based encryption, traceability, revocability, updatability.

## I. INTRODUCTION

As one of the main services in cloud computing, cloud storage servers possess the capability of powerful computation and data storage, which is viewed as the most practical and basic service in commercial application. Owning to its advantages of low costs, large amount of space, accessibility, and flexible storage management, more and more individuals and enterprises prefer to share their data including sensitive messages through the cloud server. But this common approach to achieve easy data sharing is often inconsistent with data security, since the cloud service provider can not be fully trusted and may leak the data for illegal purpose. To solve this problem, attribute-based encryption (ABE) introduced by Sahai and Waters [1] allows users to encrypt their data before outsourcing to the cloud. Moreover, ABE can also provide a kind of versatile one-to-many encryption mechanism, and thus it is regarded as a highly promising method to realize flexible access control on data sharing in cloud storage.

Until now, there are two main complementary types in ABE: ciphertext-policy attribute-based encryption (CP-ABE) [2] and key-policy attribute-based encryption (KP-ABE) [3]. In CP-ABE system, the ciphertext is tied to an access policy, and a user's secret key is related to a set of attributes. While the roles of access policies and attributes set are totally exchanged in KP-ABE. Only the attribute set

The associate editor coordinating the review of this manuscript and approving it for publication was Muhammad Imran Tariq.

meets the access policy, can the message be recovered from the ciphertext in ABE system. Since then, many improved ABE schemes have been studied in [4]–[12], aiming at more highly expressive access structures, better efficiency, and multi-authority construction.

### A. MOTIVATIONS

Although significant progress has been made in ABE, there are still some fundamental and major challenges to be solved, which could impede the broad applications. In this paper, we mainly discuss the following problems.

#### 1) MALICIOUS USER TRACING

In the conventional ABE system (CP-ABE as an example), the secret keys are defined over the multiple-user shared descriptive attributes. Therefore, the users with the same attributes can have the same decryption privileges. As a secret key is only bound to a user's attributes rather than his identity information, it also brings the problem that the secret keys are not traceable: the original key owner cannot be found through the leaked key. When a malicious user intentionally leaks his partial or entire secret key to some third party for financial profits, he has no risk of being caught.

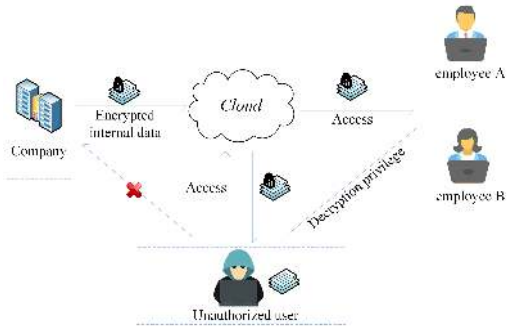


**FIGURE 1.** Malicious key leaking model in company's cloud storage system.

Given an example as shown in Fig.1, consider the commercial application scenario that a company shares their internal data by using a cloud storage system. In the company, a large amount of data will be encrypted under certain access policies through an applicable CP-ABE scheme such as [5], [6]. Suppose that there exists an important document encrypted under the access policy of {Product Development Department **AND** Production Engineer}. An employee *Alice* possesses the attribute set {Junior, Product Development Department, Production Engineer}, and another employee *Bob*'s attributes are described as {Higher, Product Development Department, Production Engineer}. Thus, both *Alice* and *Bob* can derive a decryption key for attribute set {Product Development Department, Production Engineer}. If there exists an employee belonging to the company's competitor that could recover the document, the company will suffer serious financial losses. Then, the question is who leaks the decryption privilege, *Alice* or *Bob*?

#### 2) MALICIOUS USER REVOCATION AND CIPHERTEXT UPDATE

In some practical commercial application, especially considering the company cloud storage system we mentioned above, it is essential to prevent the malicious employee from continuing to leak their decryption privilege after he was caught. That is to say, an effective revocation mechanism needs to be provided to revoke the access rights of the malicious user. And as far as we know, there are many literatures that have focused on the revocable attribute-based encryption. However, in 2012, Sahai et al. told an amazing story in [25]. Informally speaking, after the malicious employee is terminated and has his access right revoked, he can still penetrate the cloud storage server and decrypt the past ciphertext (previous unread) stored in cloud, since this employee had the insider knowledge of the company's system and retained his old key. Therefore, in order to achieve a valid revocation, it is required that the cloud service provider can only use the public information to update the ''old'' ciphertext to obtain a ''new'' ciphertext without access to any sensitive data, which avoids the decryption and then re-encryption.

### B. RELATED WORKS

Up to date, many encryption schemes against key abuse have been studied in [13]–[20]. Hinek *et al.*'s scheme [13] and Li *et al.*'s scheme [14] can only support ''**AND**'' gate with wildcard, thus both of them can not achieve highly expressive and flexible access control. And Yu *et al.*'s scheme [15] was a KP-ABE system. In KP-ABE, the data owners need to compare the recipient's access structure to formulate an attribute set for the data, which is not suitable in the practical applications. In 2013, Liu *et al.* [16] introduced the first flexible white-box traceable CP-ABE scheme supporting any monotonous access structure, which can trace the malicious users that reveal their secret keys/modified secret keys to the unauthorized third party. After that, Ning *et al.* [17], [18] gave two CP-ABE schemes with the function of traceability, which can also achieve the construction of large attribute universe and flexible access policies, respectively. More recently, Jiang *et al.* [19] proposed a provably secure and traceable CP-ABE scheme that can against ''key-delegation abuse'' in fog computing. And Yu *et al.* [20] gave a traceable and undeniable CP-ABE, which introduce a public auditor that can judge whether the traced malicious user is innocent or not. But even if the malicious users are traced, these schemes [16]–[20] can not revoke them from the cryptosystem and can not prevent the malicious users from continuing to leak their secret keys.

A valid revocation mechanism is crucial for the applied cryptosystems. In 2008, Boldyreva *et al.* [21] proposed the first scalable and efficient revocable identity-based encryption scheme by using a binary tree data structure. Then an advanced version that against decryption key exposure was

constructed by Seo and Emura [22]. After that, in order to obtain an efficient and secure revocation mechanisms, a series of works have been studied in both IBE and ABE setting in [23]–[31]. Especially, Hur and Noh [26] provided an elegant technique for fine-grained level revocation by using selective group key distribution method. Meanwhile, they also introduced a proxy-server that can update the ciphertext outsourced in the cloud to prevent the revoked users from decrypting the past ciphertext. Then, by making full use of the concept of attribute group, Li *et al.* [27] put forward a user collusion-resistant CP-ABE scheme supporting attribute-level revocation and ciphertext update. And to protect the confidentiality of the ''old'' ciphertext, Lee *et al.* [32] introduced a new model approach called self-updatable encryption (SUE), which can realize the ciphertext update by using time-evolution mechanism. In their construction, a ciphertext with time $t$ can be updated to a new valid ciphertext with the next time $t+1$ without accessing any sensitive data. Then, in an improved version [33], they extended their SUE scheme to support a time interval in ciphertexts, which is defined as time-interval SUE (TI-SUE). In the new scheme, the ciphertext at the time interval $[t_L, t_R]$ can be decrypted by a secret key with a time $t \in [t_L, t_R]$.

Aiming at tracing and revoking the malicious users from the cryptosystem, Liu *et al.* [34] and Ning *et al.* [35] gave two types of traceable and revocable CP-ABE, which can support large universe and short ciphertext size, respectively. However, their schemes can not execute the ciphertext update algorithm, which means that they can not protect the confidentiality of the past data. More recently, Lian *et al.* [36] provided a traceable and revocable CP-ABE by combining the SUE mechanism. Their construction required a time-based key update phase to realize the revocation of users, but can not revoke the user instantly. Then, Wang *et al.* [37] constructed a CP-ABE scheme with white-box traceability and attribute-level revocation. But in their building, if a revoked user who obtains the updated ciphertext, he can recover the message by skipping the normal decryption process. Thus they can not realize a valid revocation and provide the forward security.

### C. OUR CONTRIBUTION

As far as we know, most of the solutions failed to solve the above problems in the previous related work. And in this paper, we construct a novelly traceable, revocable, updatable, and expressive CP-ABE scheme (TRUE-CPABE). The main advantages of the proposed scheme are demonstrated as:

- ***White-Box Traceability***: It means that the malicious user who leaks his/her secret key to an unauthorized party will be find out. In our system, each user's secret key is embedded with a ''fixed point'' that closely related to his identity, which can not be changed by the user. When a malicious user delegate his partial or entire secret key to a third party, he will be caught through the ''fixed point'' by the tracing algorithm.
- ***Traitor Revocation***: We assign each system user a unique and random identifier, which is used to finish the decryption process. If a malicious user is caught, his identifier will be disabled. Thus he will be effectively revoked from the cryptosystem.
- ***Ciphertext update***: To protect the data confidentiality and prevent the malicious users from accessing the past data, we propose a new ciphertext update method. Unlike the previous work, we separate the secret exponent into two parts in the encryption phase: one is assigned to the access structure, and the other is to the revocation list. And the ciphertext components related to the access structure are not needed to be update when the malicious user has caught, which greatly simplifies the process of ciphertext update.
- ***Forward Secrecy***: Forward secrecy means that any users revoked from the system should not be authorized to access the subsequent ciphertext. Since the revoked user's identifier is disabled, he can not successfully complete the decryption process. Thus the newly encrypted data cannot be obtained by the revoked users.

### D. OUTLINE

In Section 2, some necessary background information will be given. And the formal definition of TRUE-CPABE and security models are developed in Section 3. Then the specific construction is clearly described in Section 4 and the security proof is shown in Section 5. In Section 6, the comparisons of theoretical performance and functionalities with some related works are provided. At last of this paper, we make a conclusion in Section 7.

## II. PRELIMINARIES

### A. ACCESS STRUCTURE

Let $\mathcal{P} = \{P_1, P_2, \ldots, P_n\}$ be a set of parties. A monotonic collection $\mathbb{A} \subseteq 2^{\mathcal{P}}$ is defined as: for $\forall\ B,\ C$, if $B \in \mathbb{A}$ and $B \subseteq C$, then $C \in \mathbb{A}$. And an access structure [2] denoted by $\mathbb{A}$ (respectively, monotone access structure) is a collection (respectively, monotone collection) of non-empty subsets of $\mathcal{P}$, i.e., $\mathbb{A} \subseteq 2^{\mathcal{P}} \setminus \{\emptyset\}$. A set contained in $\mathbb{A}$ is called as an authorized set, otherwise, it is called as unauthorized set. In ABE, the parties in $\mathcal{P}$ are replaced by attributes. And in this article, we only focus our attention on monotonic access structures.

### B. BILINEAR MAP

Let $\mathbb{G}$ and $\mathbb{G}_T$ be two multiplicative cyclic groups with the order of prime number $p$, and $g$ be a random generator of $\mathbb{G}$. A bilinear map $e$ [39]: $\mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ possesses the properties listed below:

- ***Bilinearity***: $\forall\ h, f \in \mathbb{G},\ u, v \in \mathbb{Z}_p,\ e(h^u, f^v) = e(h, f)^{uv} = e(h, f)^{vu}$;
- ***Non-degeneracy***: $e(g, g) \neq 1$;
- ***Computability***: $\forall\ h, f \in \mathbb{G}$, $e(h, f)$ can be calculated efficiently in polynomial time.

### C. LINEAR SECRET SHARING SCHEME(LSSS)

According to Beimel' scheme [40], any access structure $\mathbb{A}$ can be realized by a LSSS $(\mathcal{M}, \rho)$, where $\mathcal{M}$ is the share-generating matrix with the size of $\ell \times n$ and $\rho$ labels each row of $\mathcal{M}$ into an attribute $\rho(i)$ ($\rho$ is an injective function). In this paper, any access structure $\mathbb{A}$ used in our system will be represented by a LSSS, which consists of the following two algorithms:

- ***Share***$((\mathcal{M}, \rho), s)$: This algorithm can share a secret value $s \in \mathbb{Z}_p$ ($p$ is a large prime) to attributes. Set a vector $\vec{v} = (s, \upsilon_2, \ldots, \upsilon_n)^\top$ of length $n$, where $s$ is the secret value to be shared and $\upsilon_2, \ldots, \upsilon_n$ are chosen from $\mathbb{Z}_p$ randomly. Define $\mathcal{M}_i$ as the $i$-th row of $\mathcal{M}$ and compute the $i$-th part of secret shares as $\lambda_i = \mathcal{M}_i \cdot \vec{v}$ that belongs to the attribute $\rho(i)$.
- ***Reconstruction***$(\lambda_1, \ldots \lambda_\ell, (\mathcal{M}, \rho))$: This algorithm is able to recover $s$ from $\lambda_1, \ldots \lambda_\ell$. Define $S \in \mathbb{A}$ as any authorized set, and $I = \{i : \rho(i) \in S\} \subseteq \{1, 2, \ldots, \ell\}$. Then a set of coefficients $\{\omega_i \in \mathbb{Z}_p | i \in I\}$ can be computed such that $\sum_{i \in I} \omega_i \mathcal{M}_i = (1, 0, \ldots, 0)$, and we can reconstructure $s = \sum_{i \in I} \omega_i \lambda_i$.

### D. COMPLEXITY ASSUMPTIONS

In this part, we will briefly recall the $l$-Strong Diffie-Hellman ($l$-SDH) assumption and the decisional $q$-Bilinear Diffie-Hellman Exponent ($q$-BDHE) assumption.

*Assumption 1 (($l$-SDH) [41]):* Let $\mathbb{G}$ be a bilinear group with the order of prime number $p$, and $g$ be a random generator of $\mathbb{G}$. The $l$-SDH challenge problem is: an algorithm $\mathcal{A}$ takes as input an ($l$+1)-tuple $(g, g^z, g^{z^2}, \ldots, g^{z^l})$, and outputs a pair $(c, g^{1/(z+c)}) \in \mathbb{Z}_p \times \mathbb{G}$.

The algorithm $\mathcal{A}$ has advantage $\varepsilon$ in solving the $l$-SDH challenge problem if

$$\left| \Pr[\mathcal{A}(g, g^z, g^{z^2}, \ldots, g^{z^l}) = (c, g^{1/z+c})] \right| \geqslant \varepsilon,$$

where the probability is over the random choice of $z$ in $\mathbb{Z}_p^*$ and the random bits consumed by $\mathcal{A}$.

*Definition 1:* The $l$-SDH assumption holds if the $l$-SDH challenge problem can not be solved by any polynomial-time algorithm $\mathcal{A}$ who has at least non-negligible advantage.

*Assumption 2 (($q$-BDHE) [5]):* Let $\mathbb{G}$ and $\mathbb{G}_T$ be two multiplicative cyclic groups with the order of prime number $p$, and $g$ be a random generator of $\mathbb{G}$. Define a bilinear map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$. The $q$-BDHE challenge problem is : given

$$\vec{y} = (g, g^s, g^d, \ldots, g^{d^q}, g^{d^{q+2}}, \ldots, g^{d^{2q}}), \quad \text{where } d, s \in \mathbb{Z}_p^*$$

An algorithm $\mathcal{A}$ has difficult in distinguishing $e(g, g)^{d^{q+1}s} \in \mathbb{G}_T$ from a random element $Z$ in $\mathbb{G}_T$.

With the output of $\{0, 1\}$, $\mathcal{A}$ solves the $q$-BDHE challenge problem with advantage $\varepsilon$ if:

$$\left| \Pr[\mathcal{A}(\vec{y}, W = e(g, g)^{d^{q+1}s}) = 0] - \Pr[\mathcal{A}(\vec{y}, W = Z) = 0] \right| \geqslant \varepsilon.$$

*Definition 2:* The $q$-BDHE assumption holds if the $q$-BDHE challenge problem can not be solved by any polynomial-time algorithm $\mathcal{A}$ who has at least non-negligible advantage.

### E. SYSTEM DESCRIPTION

In order to make our system more clearly, a framework of the proposed TRUE-CPABE scheme is given in Fig.2, which contains four entities:

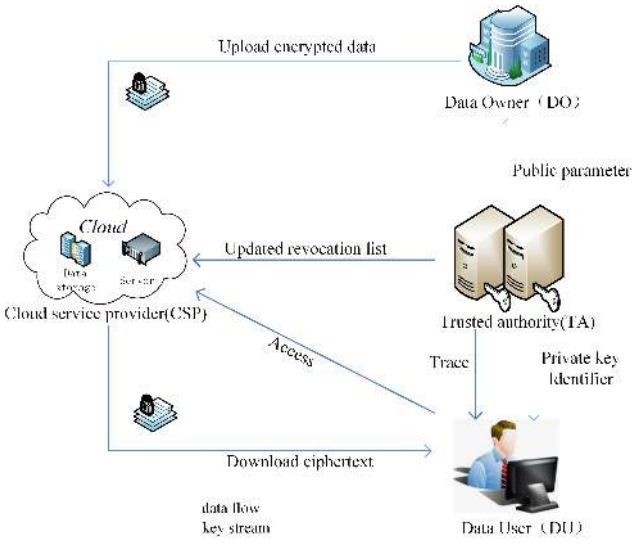


**FIGURE 2.** System model of our TRUE-CPABE scheme.

- ***Trusted Authority***(TA): TA is fully trusted and sets public parameters and master secret key for the whole system. Moreover, it takes the charge of issuing secret keys, tracing malicious users, and revoking traitors. In our construction, TA also maintains an identity table and a public revocation list.
- ***Data Owner***(DO): It is a client that wants to share his data to a specific group of users. And DO is responsible for defining an access structure, encrypting the data, and outsourcing the ciphertext to the cloud.
- ***Data User***(DU): DU can access the ciphertext outsourced in the cloud. But the message can be successfully recovered when and only when DU is not in the revocation list and possesses the authorized attribute set.
- ***Cloud Service Provider***(CSP): CSP is a service provider who is honest-but-curious, which means that CSP will honestly execute each authorization request, but get as much information as possible from the process and results. And when the revocation list is changed, CSP can update the ciphertext outsourced in the cloud.

### F. FORMAL DEFINITION OF TRUE-CPABE SCHEME

In our construction, a TRUE-CPABE scheme consists of the following six algorithms:

- ***Setup***$(\lambda, U) \to (PP, MSK)$: This algorithm is performed by TA. Taking a security parameter $\lambda$ and an

attribute universe $U$ as input, TA outputs the public parameters $PP$ and a master secret key $MSK$. Moreover, TA also initializes an identity table $T = \emptyset$ and maintains a public revocation list $R$.

- ***KeyGen***$(MSK, PP, ID, S) \rightarrow SK$: Taking as input the public parameters $PP$, the master secret key $MSK$, a user's identity $ID$, and an attribute set $S$, TA generates an identifier $id$ and a corresponding secret key $SK$. Then TA sends $id$ and $SK$ to the user via a secret channel, and adds $(ID, id)$ into $T$.
- ***Encrypt***$(PP, m, \mathbb{A}, R) \rightarrow CT$: On input the public parameters $PP$, a message $m$, an access structure $\mathbb{A}$, and the current revocation list $R$, DO performs this algorithm to generate a ciphertext $CT$.
- ***Decrypt***$(PP, id, SK, CT) \rightarrow m$ or $\perp$: Taking as input the public parameters $PP$, a user's identifier $id$, the secret key $SK$, as well as a ciphertext $CT$, DU recovers $m$ if $id$ is not contained in $R$ and $S \in \mathbb{A}$. Otherwise, output a failure symbol $\perp$ .
- ***Trace***$(PP, T, SK) \rightarrow (ID, id)$ or $\perp$: With the input of the public parameters $PP$, the identity table $T$, and a suspected secret key $SK$, TA first checks whether the $SK$ can pass the key sanity check. If $SK$ is well-formed ($SK$ can pass the check successfully), this algorithm outputs a pair of $(ID, id)$ and updates $R$ to $R'$, otherwise outputs a failure symbol "$\perp$".
- ***CTUpdate***$(PP, CT, R') \rightarrow CT'$: By inputting the public parameters $PP$, the original ciphertext $CT$, the updated revocation list $R'$, CSP outputs a new valid ciphertext $CT'$.

### G. TRACEABILITY MODEL

The definition of traceability of the TRUE-CPABE scheme is formulated by a challenge game between an adversary $\mathcal{A}$ and a simulator $\mathcal{B}$, which is obtained by use of Liu et al.'s scheme [16].

- ***Init***: $\mathcal{B}$ first executes ***Setup*** algorithm and returns the public parameters $PP$ to $\mathcal{A}$, then initializes an identity table $T = \emptyset$.
- ***Key Query***: $\mathcal{A}$ queries $\mathcal{B}$ for at most $q$ secret keys corresponding to a set of tuples $(ID_1, S_1), \ldots, (ID_q, S_q)$, where $ID_i \in R^*$ or $S_i \notin \mathbb{A}^*$, $i = 1, \ldots, q$. Then $\mathcal{B}$ performs the ***Keygen*** algorithm and gives the results to $\mathcal{A}$.
- ***Key Forgery***: In this phase, $\mathcal{A}$ outputs a secret key $SK^*$. If $Trace(PP, T, SK^*) \neq \perp$ and $Trace(PP, T, SK^*) \notin \{(ID_i, id_i), \ldots, (ID_q, id_q)\}$, then $\mathcal{A}$ will win the game. The advantage of $\mathcal{A}$ in the challenge game is:

$$Adv_{\mathcal{A}} = \Pr\Big[Trace(PP, T, SK^*) \notin \{\perp, (ID_i, id_i), \ldots, (ID_q, id_q)\}\Big].$$

***Definition 3***: If there is no polynomial-time adversary that wins the above game with at least a non-negligible advantage, then our TRUE-CPABE scheme is fully traceable.

### H. IND-CPA SECURITY MODEL

In our construction, the original ciphertext consists of two parts: the first part is tied to an access structure, and the second part is connected to the public revocation list. And when the revocation occurs, the second part of the ciphertext will be updated to avoid the revoked users accessing the past data. However, in our system, the updated ciphertext is distributed identically with the original ciphertext. Thus we only consider the sematic security of the original ciphertext.

The IND-CPA security [38] model under selective access policies attacks of our TRUE-CPABE scheme is defined by a challenge game between an adversary $\mathcal{A}$ and a simulator $\mathcal{B}$. The process of the game is given as:

- ***Init***: $\mathcal{A}$ picks $\mathbb{A}^* = (\mathcal{M}^*, \rho^*)$ ($\mathcal{M}^*$ is an $\ell^* \times n^*$ matrix with $n^* \leq q$) as the challenged access structure and $R^*$ as the challenged revocation list, then sends them to $\mathcal{B}$.
- ***Setup***: After receiving $\mathbb{A}^*$ and $R^*$, $\mathcal{B}$ returns the public parameters $PP$ to $\mathcal{A}$ by running the ***Setup*** algorithm, then initializes an identity table $T = R^*$.
- ***Phase 1***: $\mathcal{A}$ makes identifier and secret key queries corresponding to a set of tuples $(ID_1, S_1), \ldots, (ID_q, S_q)$.
  - If $S_i \in \mathbb{A}^*$ and $ID_i \notin R^*$, $i = 1, \ldots, q$, then abort.
  - If $S_i \notin \mathbb{A}^*$ or $ID_i \in R^*$, $i = 1, \ldots, q$, $\mathcal{B}$ generates an identifier and a secret key, then sends them to $\mathcal{A}$.
- ***Challenge***: $\mathcal{A}$ declares two messages $m_0$ and $m_1$ with the equal length. Then $\mathcal{B}$ chooses a random coin value $\sigma \in \{0, 1\}$ and runs ***Encrypt***$(PP, m_\sigma, \mathbb{A}^*, R^*)$ to obtain the challenged ciphertext $CT^*$ and returns it to $\mathcal{A}$.
- ***Phase 2***: The same as *Phase 1*.
- ***Guess***: $\mathcal{A}$ returns a guess $\sigma' \in \{0, 1\}$ of $\sigma$. If $\sigma' = \sigma$, $\mathcal{A}$ wins the challenge game.

Define the advantage of $\mathcal{A}$ in the security game as:

$$Adv_{\mathcal{A}} = \left| \Pr[\sigma' = \sigma] - \frac{1}{2} \right|.$$

*Definition 4:* If all polynomial-time adversaries have at most negligible advantage in the security game, then our TRUE-CPABE scheme is IND-CPA secure under the selective access policy attacks.

## III. OUR CONSTRUCTION

In this part, we will give the specific construction about TRUE-CPABE scheme based on Liu *et al.*'s scheme [16] and Lewko *et al.*'s scheme [24]. We use a simple revocation technique modified from the first method of Water's revocation systems [24] to prevent the malicious users from continuing to decrypt the ciphertext after they are caught.

To finish our scheme, each user whose identity is represented by $ID \in \{0, 1\}^*$ will be assigned with a unique and random identifier $id \in \mathbb{Z}_p$ by TA. This value is used to complete the decryption process and achieve the revocation. And in our construction, the message is encrypted under a specific access structure $\mathbb{A}$ and the current revocation list $R = \{(ID_1, id_1), \ldots, (ID_r, id_r)\}$. The encryption algorithm will create a secret exponent $s \in \mathbb{Z}_p$, which is used to hide the message. Then splits $s$ into $s'$ and $s''$, for $s = s' + s''$.

Moreover, randomly choose $s_k$ from $\mathbb{Z}_p$ make that $s'' = \sum_{k=1}^{r} s_k$. And a user's secret key is made of an attribute set $S$ that he possessed and his identifier $id$. Only a user's identifier $id$ is not contained in $R$ and his attribute set $S \in \mathbb{A}$, can he recover the message successfully. If a user's identifier $id = id_k$ ($k \in \{1, \ldots, r\}$), he can not incorporate the $k$-th share of $s''$ and thus is unable to decrypt the ciphertext.

- ***Setup***$(\lambda, U) \rightarrow (PP, MSK)$: Taking as input a security parameter $\lambda$ and an attribute universe $U$, this algorithm first runs the group generator algorithm $\mathcal{G}(\lambda)$ to obtain the bilinear group mapping description $GD = \{p, \mathbb{G}, \mathbb{G}_{\mathbb{T}}, e\}$ and a random generator $g$ of $\mathbb{G}$. Then this algorithm performs the following steps:
  - Select $\alpha, a \in \mathbb{Z}_p$, $\beta \in \mathbb{Z}_p^*$, and $h \in \mathbb{G}$ randomly.
  - For each attribute $x \in U$, randomly choose $U_x \in \mathbb{G}$.
  - Choose a probabilistic symmetric encryption approach $(Enc, Dec)$ [42], which encrypts $\{0,1\}^*$ to $\mathbb{Z}_p^*$. Moreover, this encryption algorithm can encrypt the same message to obtain different ciphertext each time with the symmetric key $\bar{k} \in \mathbb{Z}_p$.

  The public parameters are formed as:

$$PP = \langle g, h, h^{\frac{1}{\beta}}, g^a, h^a, e(g,g)^{\alpha}, \{U_x\}_{x \in U} \rangle.$$

  The master key is kept secretly by TA as:

$$MSK = \langle \alpha, \beta, a, \bar{k} \rangle.$$

  At last, TA initially sets an identity table $T = \emptyset$ and a public revocation list $R = \emptyset$.
- ***KeyGen***$(PP, MSK, ID, S) \rightarrow SK$: Taking as input a user's identity $ID \in \{0,1\}^*$ and an attribute set $S$, TA first selects $id \in \mathbb{Z}_p$ randomly as the user's identifier, and computes $c = Enc_{\bar{k}}(ID)$, where the value $c$ has the same distribution as a random element in $\mathbb{Z}_p^*$. Then, TA randomly picks $b, t \in \mathbb{Z}_p$, and generates a secret key $SK$ corresponding to $(ID, S)$ as follows:
  - The secret key component associated with the attribute set $S$ is generated as:

$$\langle K' = c, K = g^{\frac{\alpha}{a+c}} h^{bt}, \\ L = g^{bt}, L' = g^{abt}, \{K_x = U_x^{(a+c)bt}\}_{x \in S} \rangle.$$

  - In order to implement the function of revocation, the identifier $id$, which is used to complete the decryption precess, will be added into the secret key. The corresponding component is formed as:

$$\langle D = (g^{id} g^a)^{\beta \cdot (a+c) \cdot bt} \rangle.$$

  At last, TA adds $(ID, id)$ into $T$, and sends $id$ and $SK = \langle K', K, L, L', \{K_x\}_{x \in S}, D \rangle$ to the user via a secret channel.
- ***Encrypt***$(PP, m, \mathbb{A}, R) \rightarrow CT$: Taking as input the public parameters $PP$, a message $m \in \mathbb{G}_T$, an access structure $\mathbb{A} = (\mathcal{M}, \rho)$, and the current revocation list $R = \{(ID_1, id_1), \ldots, (ID_r, id_r)\}$, DO first chooses a secret exponent $s \in \mathbb{Z}_p$ and splits it into $s'$ and $s''$, that is $s = s' + s''$, then generates a ciphertext with the following steps:
  - Choose a vector $\vec{v} = (s', \upsilon_2, \ldots, \upsilon_n)^{\perp} \in \mathbb{Z}_p^n$ at random and calculate $\lambda_i = \mathcal{M}_i \cdot \vec{v}$ $(i = 1, \ldots, \ell)$.
  - Pick random elements $\tau_1, \ldots, \tau_\ell \in \mathbb{Z}_p$, and generate the ciphertext component corresponding to $\mathbb{A}$ as:

$$\langle C = m \cdot e(g,g)^{\alpha s}, C_0 = g^s, C_0' = g^{as}, \\ \{C_{i,1} h^{\lambda_i} U_{\rho(i)}^{-\tau_i}, C_{i,2} = g^{\tau_i}\}_{i=1}^{\ell} \rangle.$$

  - Select random elements $s_1, \ldots, s_r \in \mathbb{Z}_p$ such that $s'' = \sum_{k=1}^{r} s_k$, and compute the ciphertext components associated with $R$ as:

$$\langle \{C_{k,1}' = h^{\frac{1}{\beta} \cdot s_k}, C_{k,2}' = (h^{id_k} h^a)^{s_k}\}_{k=1}^{r} \rangle.$$

  Finally, DO uploads $CT$ to the cloud as:

$$CT = \langle C, C_0, C_0', \{C_{i,1}, C_{i,2}\}_{i=1}^{\ell}, \{C_{k,1}', C_{k,2}'\}_{k=1}^{r}, \mathbb{A}, R \rangle.$$

- ***Decrypt***$(PP, id, SK, CT) \rightarrow m$ or $\perp$: On input the public parameters $PP$, a user's identifier $id$ and secret key $SK$, as well as the ciphertext $CT$. There exist two cases:

  ***Case 1.*** If $S \notin \mathbb{A}$ or $id \in (id_1, \ldots, id_r)$, then this algorithm outputs a failure symbol $\perp$.

  ***Case 2.*** If $S \in \mathbb{A}$ and $id \notin (id_1, \ldots, id_r)$, this algorithm recovers the message $m$ by performing the following steps:
  - Set $I = \{i : \rho(i) \in S\} \subseteq \{1, \ldots, \ell\}$, and compute the coefficients $\{\omega_i \in \mathbb{Z}_p \mid i \in I\}$ such that $\sum_{i \in I} \omega_i \mathcal{M}_i = (1, 0, \ldots, 0)$, and $\sum_{i \in I} \omega_i \lambda_i = s'$.
  - Compute the following values:

$$\begin{aligned} K_1 &= e(K, C_0^{K'} \cdot C_0') \\ &= e(g^{\frac{\alpha}{a+c}} h^{bt}, g^{sc} \cdot g^{as}) \\ &= e(g,g)^{\alpha s} e(g,h)^{(a+c)bts}, \\ K_1' &= \prod_{i \in I} \Big( e(L^{K'} \cdot L', C_{i,1}) \cdot e(K_{\rho(i)}, C_{i,2}) \Big)^{\omega_i} \\ &= \prod_{i \in I} \Big( e\big(g^{(a+c)bt}, h^{\lambda_i} U_{\rho(i)}^{-\tau_i}\big) \cdot e(U_{\rho i}^{(a+c)bt}, g^{\tau_i}) \Big)^{\omega_i} \\ &= \prod_{i \in I} \Big( e(g^{(a+c)bt}, h^{\lambda_i}) \cdot e(g^{(a+c)bt}, U_{\rho(i)}^{-\tau_i}) \\ &\quad \cdot e(U_{\rho i}^{(a+c)bt}, g^{\tau_i}) \Big)^{\omega_i} \\ &= \prod_{i \in I} e(g,h)^{(a+c)bt \lambda_i \omega_i} \\ &= e(g,h)^{(a+c)bts'}, \\ K_1'' &= \frac{e\Big(D, \prod_{k=1}^{r} (C_{k,1}')^{\frac{1}{id - id_k}}\Big)}{e\Big(L^{K'} \cdot L', \prod_{k=1}^{r} (C_{k,2}')^{\frac{1}{id - id_k}}\Big)} \end{aligned}$$

$$= \frac{e\left(\left(g^{id}g^{a}\right)^{\beta\cdot(a+c)bt}, \prod_{k=1}^{r}(h^{\frac{1}{\beta}\cdot s_k})^{\frac{1}{id-id_k}}\right)}{e\left(g^{(a+c)bt}, \prod_{k=1}^{r}\left((h^{id_k}h^{a})^{s_k}\right)^{\frac{1}{id-id_k}}\right)}$$

$$= \frac{\prod_{k=1}^{r}\left(e(g^{id\cdot\beta}, h^{\frac{1}{\beta}\cdot s_k})\cdot e(g^{a\cdot\beta}, h^{\frac{1}{\beta}s_k})\right)^{(a+c)bt\cdot\frac{1}{id-id_k}}}{\prod_{k=1}^{r}\left(e(g,\ h^{id_k s_k})\cdot e(g, h^{as_k})\right)^{(a+c)bt\cdot\frac{1}{id-id_k}}}$$

$$= \prod_{k=1}^{r} e(g,h)^{(a+c)bt\cdot s_k\cdot(id-id_k)\cdot\frac{1}{id-id_k}}$$

$$= \prod_{k=1}^{r} e(g,h)^{(a+c)bt\cdot s_k}$$

$$= e(g,h)^{(a+c)bts''},$$

$$K_2 = \frac{K_1}{K_1'\cdot K_1''}$$

$$= \frac{e(g,g)^{\alpha s}\cdot e(g,h)^{(a+c)bts}}{e(g,h)^{(a+c)bts'}\cdot e(g,h)^{(a+c)bts''}}$$

$$= e(g,g)^{\alpha s}.$$

- Retrieve the message $m = \frac{C}{K_2} = \frac{m\cdot e(g,g)^{\alpha s}}{e(g,g)^{\alpha s}}$.

- ***Trace***$(PP,\ T,\ SK) \rightarrow (ID, id)$ or $\perp$: Taking as input the public parameters $PP$, the identity table $T$, and a suspicious secret key $SK$, TA first checks whether $SK$ can pass the ***Key Sanity Check***. This check ensures that $SK$ can be used in a well-formed decryption procedure.

  ***Key Sanity Check***:

$$K' \in \mathbb{Z}_p, K, L, L', K_x, D \in \mathbb{G}. \quad (1)$$

$$e(g, L') = e(g^a, L) \neq 1. \quad (2)$$

$$e(g^a\cdot g^{K'}, K) = e(g,g)^{\alpha}\cdot e(L^{K'}\cdot L', h) \neq 1. \quad (3)$$

$$\exists x \in S,\ s.t.\ e(U_x, L^{K'}\cdot L') = e(g, K_x) \neq 1. \quad (4)$$

  ***Case 1***. If $SK$ goes through the ***Key Sanity Check***, it indicates that $SK$ is well-formed. Then TA does as follows:

  - Extract $ID$ from $Dec_{\bar{k}}(K') = Dec_{\bar{k}}(Enc_{\bar{k}}(ID))$.
  - Search $ID$ from $T\backslash R$. If $ID$ is contained in $T\backslash R$, output the identity tuple $(ID,\ id)$. Otherwise, output $(ID^*,\ id^*)$ as the result, which is a special identity tuple and not in the system.
  - Update the current revocation list $R$ to

$$R' = R \cup \left\{(ID_{r+1},\ id_{r+1}) = (ID,\ id)\right\},$$

    then send $R'$ to CSP.

  ***Case 2***. If $SK$ can not go through the ***Key Sanity Check***, this algorithm returns a special failure symbol "$\perp$."

- ***CTUpdate***$(PP,\ CT,\ R') \rightarrow CT'$: Upon receiving the new revocation list $R'$, CSP updates the original ciphertext $CT$ to a new valid ciphertext $\widetilde{CT}$ that is tied to the updated revocation list $R'$ with the following steps:

  - Choose $\tilde{s} \in \mathbb{Z}_p$ randomly, then compute

$$\tilde{C} = C\cdot e(g,g)^{\alpha\tilde{s}} = m\cdot e(g,g)^{\alpha(s+\tilde{s})},$$
$$\tilde{C}_0 = C_0\cdot g^{\tilde{s}} = g^{s+\tilde{s}},$$
$$\tilde{C}_0' = C_0'\cdot g^{a\tilde{s}} = g^{a(s+\tilde{s})},$$
$$\{\tilde{C}_{i,1} = C_{i,1},\ \tilde{C}_{i,2} = C_{i,2}\}_{i\in\{1,\ldots,\ell\}}.$$

  - Pick random elements $\tilde{s}_1, \ldots \tilde{s}_{r+1} \in \mathbb{Z}_p$ such that $\tilde{s} = \sum_{k=1}^{r+1}\tilde{s}_k$, then compute the new ciphertext components related to $R'$ as:

$$\tilde{C}_{k,1}' = C_{k,1}'\cdot h^{\frac{1}{\beta}\cdot\tilde{s}_k} = h^{\frac{1}{\beta}(s_k+\tilde{s}_k)},$$

$$\tilde{C}_{k,2}' = C_{k,2}'\cdot(h^{id_k}h^{a})^{\tilde{s}_k} = (h^{id_k}h^{a})^{s_k+\tilde{s}_k},$$

$$\tilde{C}_{r+1,1} = h^{\frac{1}{\beta}\cdot\tilde{s}_{r+1}},$$

$$\tilde{C}_{r+1,2} = (h^{id}h^{a})^{\tilde{s}_{r+1}}.$$

  Finally, the updated ciphertext is

$$\widetilde{CT} = \langle\tilde{C}, \tilde{C}_0, \tilde{C}_0', \{\tilde{C_{i,1}}, \tilde{C}_{i,2}\}_{i=1}^{\ell}, \{\tilde{C}_{k,1}', \tilde{C}_{k,2}'\}_{k=1}^{r},$$
$$\{\tilde{C}_{r+1,1}, \tilde{C}_{r+1,2}\}, \mathbb{A}, R'\rangle.$$

## IV. SECURITY PROOF

### A. TRACEABILITY

In this part, the traceability of the proposed TRUE-CPABE scheme will depend on the $l$-SDH assumption. And the approach used in the proof process is analogous to that of Lemma 1 in Boneh et al.' scheme [41].

*Theorem 1:* Our TRUE-CPABE scheme is fully traceable if the $l$-SDH assumption holds and the number of key queries $q < l$.

*Proof:* Suppose that there is an adversary $\mathcal{A}$ that wins the traceability game with non-negligible advantage $\varepsilon$ in polynomial time by performing $q$ times key queries, w.l.o.g., assume that $l = q + 1$, then we will be able to construct a polynomial-time algorithm $\mathcal{B}$ that solves the $l$-SDH challenge problem with the same advantage $\varepsilon$.

Let $\mathbb{G}$ be a bilinear group with the order of prime number $p$, a bilinear map $e$: $\mathbb{G}\times\mathbb{G} \rightarrow \mathbb{G}_T$. To solve the $l$-SDH challenge problem, $\mathcal{B}$ is given an instance of $(\bar{g},\ \bar{g}^{a},\ \bar{g}^{a^2}, \ldots,\ \bar{g}^{a^l})$, where $a \in \mathbb{Z}_p^*$ and$\bar{g} \in \mathbb{G}$. And $\mathcal{B}$ is required to output a pair $(c_r,\ \omega_r) \in \mathbb{Z}_p \times \mathbb{G}$, which satisfies $\omega_r = \bar{g}^{1/(a+c_r)}$. $\mathcal{B}$ first sets $A_i = \bar{g}^{a^i}$ for $i = 1, \ldots, l$, then simulates the role of a challenger for $\mathcal{A}$ as follows:

- ***Init***: $\mathcal{B}$ chooses $q$ distinct values $c_1, \ldots, c_q \in \mathbb{Z}_p^*$ uniformly at random, and sets a polynomial $f(y)$ as:

$$f(y) = \prod_{i=1}^{q}(y + c_i) = \sum_{i=0}^{q}\alpha_i y^i,$$

  where $\alpha_0, \ldots, \alpha_q \in \mathbb{Z}_p$ are the cofficients of $f(y)$. Then $\mathcal{B}$ performs the following steps:

  - Choose $\alpha, \theta \in \mathbb{Z}_p$, $\beta \in \mathbb{Z}_p^*$ randomly.

– Compute

$$g = \prod_{i=1}^{q}(A_i)^{\alpha_i} = \bar{g}^{f(a)},$$
$$g^a = \prod_{i=0}^{q+1}(A_i)^{\alpha_{i-1}} = \bar{g}^{f(a)\cdot a},$$

and

$$h = g^{\theta},\ h^a = (g^a)^{\theta},\ h^{\frac{1}{\beta}} = (g^{\theta})^{\frac{1}{\beta}}.$$

– For each attribute $x \in U$, select a random element $u_x \in \mathbb{Z}_p$, set the attribute parameter $U_x = g^{u_x}$, then initialize an identity table $T = \emptyset$.

Finally, the simulated public parameters are formed as:

$$PP = \langle g,\ h,\ h^{\frac{1}{\beta}},\ g^a,\ h^a,\ e(g,g)^{\alpha},\ \{U_x\}_{x\in U}\rangle.$$

- ***Key Query***: At this stage, $\mathcal{A}$ submits a sequence of tuples $(ID_i, S_i)$ to $\mathcal{B}$ requesting the corresponding secret keys and identifers. Presume that it is the $i$-th query, such that $i \le q$. $\mathcal{B}$ does the following steps:
  – Randomly choose $b,\ t \in \mathbb{Z}_p$. If $ID_i \in T$, then extract $id_i$ from $T$. Otherwise, pick a random element $id_i \in \mathbb{Z}_p \backslash \{id_1, \ldots, id_{i-1}\}$, and add $(ID_i, id_i)$ into the identity table $T$.
  – Set the polynomial $f_i(y)$ as:

$$f_i(y) = \frac{f(y)}{y + c_i} = \prod_{j=1, j\neq i}^{q}(y + c_i) = \sum_{j=0}^{q-1}\beta_j y^j.$$

  – Compute

$$\sigma_i = \prod_{j=0}^{q-1}(A_j^{\beta_j}) = \bar{g}^{f_i(a)} = \bar{g}^{f(a)/(a+c_i)} = g^{1/(a+c_i)}.$$

  – Generate the secret key components as:

$$K' = c_i,\quad K = (\sigma_i)^{\alpha}(g^{\theta})^{bt},$$
$$L = g^{bt},\ L' = (g^a)^{bt},$$
$$\{K_x = (g^a \cdot g^{c_i})^{u_x bt}\}_{x\in S_i},$$
$$D = (g^{id_i} g^a)^{\beta\cdot(a+c_i)\cdot bt}.$$

Finally, $\mathcal{B}$ sends $\mathcal{A}$ the identifier $id_i$ and the secret key

$$SK_i = \langle K', K, L, L', \{K_x\}_{x\in S_i}, D\rangle.$$

- ***Key Forgery***: $\mathcal{A}$ will submit a secret key $SK^*$ to $\mathcal{B}$. Let $\xi_{\mathcal{A}}$ represent the event that $\mathcal{A}$ wins the traceability game, i.e., $SK^*$ is shaped into the form as $SK^* = \langle K', K, L, L', \{K_x\}_{x\in S_i}, D\rangle$ and passes the ***Key Sanity Check***, as well as $K' \notin \{c_1, \ldots, c_q\}$.
  – If $\xi_{\mathcal{A}}$ does not happen, it means that $\mathcal{B}$ does not get any useful information. Thus $\mathcal{B}$ chooses $(c_r, \omega_r) \in \mathbb{Z}_p \times \mathbb{G}$ randomly as a solution to the $l$-SDH challenge problem.
  – If $\xi_{\mathcal{A}}$ happens, $\mathcal{B}$ does the following steps:

1) Set the polynomial

$$f(y) = \prod_{i=1}^{q}(y + c_i) = \gamma(y)(y + K') + \gamma - 1,$$

for some polynomial $\gamma(y) = \sum_{i=0}^{q-1}(\gamma_i y^i)$ and some $\gamma - 1 \in \mathbb{Z}_p$. Since $f(y) = \prod_{i=1}^{q}(y + c_i)$, $c_i \in \mathbb{Z}_p^*$ and $K' \notin \{c_1, c_2, \ldots, c_q\}$, $f(y)$ can not be divided by $y + K'$. Then, we have $\gamma - 1 \neq 0$.

2) Assume that $L = g^{bt}$, where $b,\ t \in \mathbb{Z}_p$ are unknown. According to the equation (2) and (3) in the ***Key Sanity Check***, we can obtain

$$L' = g^{abt} \text{ and } K = g^{\frac{\alpha}{a+K'}} h^{bt}.$$

3) Compute $\frac{1}{\gamma-1}$ when $\gcd(\gamma - 1, p) = 1$, and

$$\sigma = (K/L^{\theta})^{\alpha^{-1}} = g^{\frac{1}{a+K'}} = \bar{g}^{\gamma(a)}\bar{g}^{\frac{\gamma-1}{a+K'}},$$
$$\omega_r = (\sigma \cdot \prod_{i=0}^{q-1} A_i^{-\gamma_i})^{\frac{1}{\gamma-1}} = \bar{g}^{\frac{1}{a+K'}},$$
$$c_r = K' \bmod p.$$

For $e(\bar{g}^a \cdot \bar{g}^{c_r},\ \omega_r) = e(\bar{g}^a \cdot \bar{g}^{K'},\ \bar{g}^{\frac{1}{a+K'}}) = e(\bar{g}, \bar{g})$, $(c_r, \omega_r)$ is a correct solution to the $l$-SDH challenge problem.

Let $\zeta$ denote that $(c_r,\ \omega_r)$ is a solution for the $l$-SDH challenge problem, which can be tested by whether $e(\bar{g}^a \cdot \bar{g}^{c_r}, \omega_r) = e(\bar{g}, \bar{g})$ holds. If $\xi_{\mathcal{A}}$ does not occur, $\mathcal{B}$ randomly chooses $(c_r, \omega_r) \in \mathbb{Z}_p \times \mathbb{G}$, thus $\zeta$ happens with a negligible probability, which we define as 0 for simplicity. When $\xi_{\mathcal{A}}$ happens and $\gcd(\gamma - 1, p) = 1$, $\mathcal{B}$ outputs a tuple $(c_r, \omega_r)$ satisfying $e(\bar{g}^a \cdot \bar{g}^{c_r}, \omega_r) = e(\bar{g}, \bar{g})$ with the probability as 1. The probability of $\mathcal{B}$ solving the $l$-SDH challenge problem is:

$$\begin{aligned}\Pr[\zeta] &= \Pr\left[\ \zeta \mid \overline{\mathcal{A}\ wins}\ \right] \cdot \Pr\left[\ \overline{\mathcal{A}\ wins}\ \right] \\ &\quad + \Pr\left[\ \zeta \mid \mathcal{A}\ wins \wedge \gcd(\gamma - 1, p) \neq 1\ \right] \\ &\quad \cdot \Pr[\mathcal{A}\ wins \wedge \gcd(\gamma - 1, p) \neq 1] \\ &\quad + \Pr\left[\ \zeta \mid \mathcal{A}\ wins \wedge \gcd(\gamma - 1, p) = 1\right] \\ &\quad \cdot \Pr[\mathcal{A}\ wins \wedge \gcd(\gamma - 1, p) = 1] \\ &= 0 + 0 + 1 \cdot \Pr\left[\ \mathcal{A}\ wins \wedge \gcd(\gamma - 1, p) = 1\right] \\ &= \Pr\left[\ \mathcal{A}\ wins \wedge \gcd(\gamma - 1, p) = 1\right] \\ &= \varepsilon.\end{aligned}$$

And the advantage of $\mathcal{B}$ in the $l$-SDH challenge game is

$$Adv_{\mathcal{B}} = \Pr\ [\ \zeta\ ] = \Pr\left[\ \mathcal{A}\ wins \wedge \gcd(\gamma - 1, p) = 1\right] = \varepsilon.$$

### B. IND-CPA SECURITY PROOF

In our construction, the ciphertext will be updated when the revocation list is changed. Since the updated ciphertexte distribute identically with the original ciphertext, we only consider the sematic security of the original ciphertext. In this part, we will reduce the IND-CPA security (indistinguishability of ciphertext under chosen plaintext attacks)

of our TRUE-CPABE scheme to the decisional $q$-BDHE assumption.

*Theorem 2:* Our scheme is IND-CPA secure under selective access policy attacks if the decisional $q$-BDHE assumption holds.

*Proof:* Assuming that there is a polynomial-time adversary $\mathcal{A}$ who has at least non-negligible advantage $\varepsilon$ in breaking our scheme, then a polynomial-time simulator $\mathcal{B}$ with the advantage of $\frac{\varepsilon}{2}$ can be built to solve the decisional $q$-BDHE challenge problem.

Let $\mathbb{G}$ and $\mathbb{G}_T$ be two multiplicative cyclic groups with the order of prime number $p$, a bilinear map $e$: $\mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$, and $g$ be a random generator of $\mathbb{G}$. $\mathcal{B}$ is given an instance of the decisional $q$-BDHE challenge problem as: $\vec{y} = (g, g^s, g^d, ..., g^{d^q}, g^{d^{q+2}}, ..., g^{d^{2q}})$ and $W$. If a coin flip $v = 1$, then $W = e(g, g)^{d^{q+1}s}$; otherwise, $W$ is randomly chosen from $\mathbb{G}_T$. And $\mathcal{B}$ is required to output a guess $v' \in \{0, 1\}$ of $v$.

- ***Init***: $\mathcal{A}$ defines an access structure $\mathbb{A}^* = (\mathcal{M}^*, \rho^*)$ and a revocation list $R^* = \{(ID_1^*, id_1^*), \ldots, (ID_r^*, id_r^*)\}$ to be challenged, where $\mathcal{M}^*$ is a matrix with the size of $\ell^* \times n^*$ and $n^* < q$.
- ***Setup***: After receiving $\mathbb{A}^*$ and $R^*$, $\mathcal{B}$ initially sets the identity table $T = R^*$, then simulates the public parameters as:
  - Pick $\alpha' \in \mathbb{Z}_p$ randomly, and compute $e(g, g)^{\alpha} = e(g^d, g^{d^q}) \cdot e(g, g)^{\alpha'}$, which implicity set $\alpha = \alpha' + d^{q+1}$.
  - Randomly select $a \in \mathbb{Z}_p$, $\beta \in \mathbb{Z}_p^*$, set $h = g^d$, and compute $g^a$, $h^a = (g^d)^a$, $h^{\frac{1}{\beta}} = (g^d)^{\frac{1}{\beta}}$.
  - Each attribute $x \in U$, pick $u_x \in \mathbb{Z}_p$ randomly, and calculate the attribute parameter $U_x$ as :
    1) If the function $\rho^*$ labels an $i \in \{1, 2, \ldots, \ell^*\}$ into $\rho^*(i) = x$, set
    $$U_x = g^{u_x}(g^d)^{\mathcal{M}^*_{i,1}}(g^{d^2})^{\mathcal{M}^*_{i,2}} \ldots (g^{d^{n^*}})^{\mathcal{M}^*_{i,n^*}}.$$
    2) If there is not an $i \in \{1, 2, \ldots, \ell^*\}$ marked with $\rho^*(i) = x$, set $U_x = g^{u_x}$.

  Finally, the public parameters are formulated as:
  $$PP = \langle g, h, h^{\frac{1}{\beta}}, g^a, h^a, e(g, g)^{\alpha}, \{U_x\}_{x \in U} \rangle.$$
- ***Phase 1***: At this stage, $\mathcal{A}$ submits a sequence of tuples $(ID_1, S_1), \ldots, (ID_q, S_q)$ to $\mathcal{B}$ asking for the corresponding identifers and secret keys. $\mathcal{B}$ responses in the following ways:
  ***Case 1***. If $S_j \in \mathbb{A}^*$ and $ID_j \notin \{ID_1^*, \ldots, ID_r^*\}$, $j \in \{1, \ldots, q\}$, then abort.
  ***Case 2***. If $S_j \in \mathbb{A}^*$ and $ID_j \in \{ID_1^*, \ldots, ID_r^*\}$, $j \in \{1, \ldots, q\}$, then randomly choose $b, c \in \mathbb{Z}_p$ and perform the following steps by implicity setting
  $$t = -\frac{d^q}{b(a+c)} + \frac{d^{q-1}}{b(a+c)} \cdot \frac{\mathcal{M}^*_{i,1}}{\mathcal{M}^*_{i,2}}.$$
  - Compute $K', K, L, L', K_x$ as:
  $$K' = c,$$
  $$K = (g^{\alpha'})^{\frac{1}{a+c}}(g^{d^q})^{\frac{\mathcal{M}^*_{i,1}}{(a+c)\mathcal{M}^*_{i,2}}} = g^{\frac{\alpha}{a+c}}h^{bt},$$
  $$L = \left((g^{d^q})^{\frac{1}{a+c}}\right)^{-1}(g^{d^{q-1}})^{\frac{\mathcal{M}^*_{i,1}}{(a+c)\mathcal{M}^*_{i,2}}} = g^{bt},$$
  $$L' = (L)^a = g^{abt},$$
  $$K_x = \left(\left(g^{d^q}\right)^{u_x}\right)^{-1}\left(g^{d^{q-1}}\right)^{u_x}\left(\prod_{j=2,\ldots,n^*}\left(g^{d^{q+j}}\right)^{\mathcal{M}^*_{i,j}}\right)^{-1} \cdot \prod_{j=1,\ldots,n^*, j\neq 2}\left(\left(g^{d^{q+j-1}}\right)^{\mathcal{M}^*_{i,j}}\right)^{\frac{\mathcal{M}^*_{i,1}}{\mathcal{M}^*_{i,2}}} = U_x^{(a+c)bt}.$$
  - As for $ID \in \{ID_1^*, \ldots, ID_r^*\}$ (there exists a $k \in \{1, \ldots, r\}$ such that $ID = ID_k^*$), the simulator $\mathcal{B}$ extracts the identifier $id = id_k^*$ from the challenge revocation list $R^*$. Then $\mathcal{B}$ calculates $D$ as:
  $$D = \left(\left(g^{d^q}\right)^{-1}\left(g^{d^{q-1}}\right)^{\frac{\mathcal{M}^*_{i,1}}{\mathcal{M}^*_{i,2}}}\right)^{\beta \cdot (id+a)} = (g^{id}g^a)^{\beta \cdot (a+c) \cdot bt}.$$

***Case 3***. If $S_j \notin \mathbb{A}^*$ and $ID_j \in \{ID_1^*, \ldots, ID_r^*\}$, $j \in \{1, \ldots, q\}$, $\mathcal{B}$ generates an identifier and a secret key as follows:

- Find a column vector $\vec{w} = (w_1, \ldots, w_{n^*}) \in \mathbb{Z}_p^{n^*}$ with the first component $w_1 = -1$. For all $i$ such that $\rho^*(i) \in S$, then $\mathcal{M}_i^* \cdot \vec{w} = 0$. According to the definition of the LSSS, such a vector must exist [40].
- Randomly choose $c, b, \theta \in \mathbb{Z}_p$, and set $t$ as:
$$t = \frac{1}{b(a+c)}(\theta + w_1 d^q + w_2 d^{q-1} + \cdots + w_{n^*} d^{q-n^*+1}).$$
- Generate $K', L, L', K$ as:
$$K' = c,$$
$$K = \left(g^{\alpha'}(g^d)^{\theta}\prod_{i=2,\ldots,n^*}\left(g^{d^{q+2-i}}\right)^{w_i}\right)^{\frac{1}{a+c}} = g^{\frac{\alpha}{a+c}}h^{bt},$$
$$L = \left(g^{\theta}\prod_{i=1,\ldots,n^*}\left(g^{d^{q+1-i}}\right)^{w_i}\right)^{\frac{1}{a+c}} = g^{bt},$$
$$L' = L^a = g^{abt}.$$
- Generate $\{K_x\}_{x \in S}$ in the following two types:
  ***Type 1*** : If there is not an $i \in \{1, \ldots, \ell^*\}$ such that $\rho^*(i) = x (x \in S)$, set $K_x$ as:
  $$K_x = (g^{u_x})^{(a+c)bt} = L^{(a+c)u_x} = U_x^{(a+c)bt}.$$

***Type 2*** : If $\rho^*$ labels an $i \in \{1, \ldots, \ell^*\}$ into the attribute $\rho^*(i) = x (x \in S)$, compute $K_x$ as:

$$\begin{aligned} K_x &= (g^{u_x} g^{d\mathcal{M}^*_{i,1}} g^{d^2\mathcal{M}^*_{i,2}} \ldots g^{d^{n^*}\mathcal{M}^*_{i,n^*}})^{(a+c)bt} \\ &= L^{(a+c)u_x} \Bigg( \prod_{j=1,\ldots,n^*} \Big( (g^{d^j})^{\theta} \\ &\quad \cdot \prod_{k=1,\ldots,n^*, k\neq j} (g^{d^{q+1+j-k}}) \Big)^{w_i} \Bigg) \\ &= U_x^{(a+c)bt}. \end{aligned}$$

– Compute $D$ as ***Case 2***.

***Case 4***. If $S_j \notin \mathbb{A}^*$ and $ID_j \notin \{ID^*_1, \ldots, ID^*_r\}$, $j \in \{1, \ldots, q\}$, $\mathcal{B}$ computes $K', K, L, L', K_x$ as ***Case 3***. As for the component $D$, $\mathcal{B}$ first checks whether $ID$ is contained in the identity table $T$. If $ID \in T$, extract $id$ from $T$. Otherwise, choose a random element $id \in \mathbb{Z}_p$, which is not in $T$, and add $(ID, id)$ into $T$. Then compute $D$ as:

$$\begin{aligned} D &= \left( g^{\theta} \prod_{i=1,\ldots,n^*} \left( g^{d^{q+1-i}} \right)^{\omega_i} \right)^{\beta(id+a)} \\ &= (g^{id} g^{a})^{\beta\cdot(a+c)\cdot bt}. \end{aligned}$$

- ***Challenge***: $\mathcal{A}$ declares two messages $m_0, m_1 \in \mathbb{G}_T$ with the equal length, then $\mathcal{B}$ sets the challenge ciphertext as follows:
  – Randomly choose $\bar{s} \in \mathbb{Z}_p$, a coin flip $\sigma \in \{0, 1\}$, then compute

$$\begin{aligned} C &= m_\sigma \cdot W \cdot e(g^d, g^{d^q})^{\bar{s}} \cdot e(g^s \cdot g^{\bar{s}}, g^{\alpha'}), \\ C_0 &= g^s \cdot g^{\bar{s}}, \\ C'_0 &= (g^s)^a \cdot (g^a)^{\bar{s}}. \end{aligned}$$

  – Pick $y_2, \ldots, y_{n^*} \in \mathbb{Z}^*_p$, and $\tau'_1, \ldots, \tau'_{\ell^*} \in \mathbb{Z}_p$ randomly, and compute $C_{i,1}, C_{i,2}, (i = 1, \ldots, \ell^*)$ as:

$$\begin{aligned} C_{i,1} &= (g^s)^{-u_{\rho^*(i)}} g^{u_{\rho^*(i)}\tau'_i} \cdot \prod_{j=2,\ldots,n^*} (g^d)^{y_j \mathcal{M}^*_{i,j}} \\ &\quad \cdot \prod_{j=1,\ldots,n^*} (g^{d^j})^{\mathcal{M}^*_{i,j}\cdot\tau'_i}, \\ C_{i,2} &= (g^s) \cdot g^{-\tau'_i}. \end{aligned}$$

  which implicity set $\tau_i = s - \tau'_i$, $i \in \{1, \ldots, \ell^*\}$, and $\vec{v} = (s, sd + y_2, sd^2 + y_3, \ldots, sd^{n^*-1} + y_{n^*})$.
  – Randomly choose $\bar{s}_1, \ldots, \bar{s}_r \in \mathbb{Z}_p$, which makes that $\bar{s} = \sum_{k=1}^{r} \bar{s}_k$. Then calculate $C'_{k,1}$ and $C'_{k,2}, (k = 1 \ldots, r)$ as:

$$C'_{k,1} = (g^d)^{\frac{1}{\beta}\cdot\bar{s}_k}, \; C'_{k,2} = (g^d)^{(id^*_k + a)\cdot\bar{s}_k}.$$

  Finally, $\mathcal{B}$ sends the challenged ciphertext $CT^*$ to $\mathcal{A}$ as:

$$CT^* = \langle C, C_0, C'_0, \{C_{i,1}, C_{i,2}\}_{i=1}^{\ell^*}, \{C'_{k,1}, C'_{k,2}\}_{i=1}^{r} \rangle.$$

- ***Phase 2***: The same as ***Phase 1***.
- ***Guess***: At last of the game, $\mathcal{A}$ returns $\sigma'$ as a guess of $\sigma$. If $\sigma' = \sigma$, $\mathcal{B}$ outputs $\nu' = 1$, which indicates that $W = e(g, g)^{d^{q+1}s}$. Otherwise, $\mathcal{B}$ outputs $\nu' = 0$, which means that $W$ is randomly chosen from $\mathbb{G}_T$.

As can be seen from the above game, the public parameters and the results of key queries in our simulation are identical to the real system.

When $\nu = 0$, $W$ is randomly chosen from $\mathbb{G}_T$, thus the message $m_\sigma$ is completely hidden from the adversary $\mathcal{A}$. So $\mathcal{A}$ wins the game with the probability $\Pr[\sigma' = \sigma | \nu = 0] = \frac{1}{2}$. $\mathcal{B}$ outputs $\nu' = 0$ when $\sigma' \neq \sigma$, and $\Pr[\nu' = \nu | \nu = 0] = \frac{1}{2}$.

When $\nu = 1$, $\mathcal{B}$ gives a prefect simulation of the challenged ciphertext. Suppose that the advantage of $\mathcal{A}$ is $\varepsilon$ in breaking the system, $\mathcal{B}$ guesses $\nu' = 1$ when $\sigma' = \sigma$, and then we have $\Pr[\nu' = \nu | \nu = 1] = \frac{1}{2} + \varepsilon$.

The advantage of $\mathcal{B}$ in solving the decisional $q$-BDHE challenge problem is:

$$\begin{aligned} Adv_{\mathcal{B}} &= |\Pr[\nu' = \nu | \nu = 0] \cdot \Pr[\nu = 0] \\ &\quad + \Pr[\nu' = \nu | \nu = 1] \cdot \Pr[\nu = 1] - \frac{1}{2}| \\ &= \left| \frac{1}{2} \cdot \frac{1}{2} + \frac{1}{2} \cdot (\frac{1}{2} + \varepsilon) - \frac{1}{2} \right| = \frac{1}{2}\varepsilon. \end{aligned}$$

## V. PERFORMANCE ANALYSIS

For convenience, the notations used in the following description is introduced in Tab. 1. In this section, two tables will be given to show the theoretical comparisons of functionality and efficiency between our approach and some related works.

**TABLE 1. Notations.**

| Notations | Terms |
|---|---|
| $\lvert\mathbb{Z}_p\rvert$ | The bitstring length of an element in $\mathbb{Z}_p$ |
| $\lvert\mathbb{G}\rvert$ | The bitstring length of an element in $\mathbb{G}$ |
| $\lvert\mathbb{G}_T\rvert$ | The bitstring length of an element in $\mathbb{G}_T$ |
| $u$ | The maximum number of users in the system |
| $E$ | Time cost of an exponentiations (*1.882ms*) |
| $P$ | Time cost of bilinear pairing (*16.064ms*) |
| $N_x$ | The maximum number of nodes in minimum cover set of attribute group |
| $\ell$ | The maximum number of rows in the share generation matrix |
| $n_I$ | The maximum number of attributes satisfying the access policy |
| $n_S$ | The maximum number of attributes in the decryption key |
| $r$ | The maximum number of revoked users |

**TABLE 2. Functionality comparisions.**

| Schemes | Instant revocation | Traceability | Ciphertext Update | Forward Secrecy |
|---|---|---|---|---|
| Ning et al. [36] | √ | √ | - | - |
| Lian et al. [37] | × | √ | √ | √ |
| Wang et al. [38] | √ | √ | √ | × |
| Ours | √ | √ | √ | √ |

As shown in Tab. 2, compared with the recent existing approaches, our TRUE-CPABE scheme can achieve three functionalities: white-box tracing, traitor revocation, and ciphertext update, and has the characteristic of forward secrecy, which enable the proposed scheme more suitable for practical and complexity commercial application.

Tab. 3 gives a comprehensive comparisons of the storage cost and the computation overhead of our scheme with [36]

**TABLE 3.** The system efficiency comparison.

| | Lian et al. [37] | Wang et al. [38] | Ours |
|---|---|---|---|
| Size of CT | $\lvert\mathbb{Z}_p\rvert + \left(\frac{\log\mathcal{U}\cdot(\log\mathcal{U}+1)}{2}(4+n_s)\right)\lvert\mathbb{G}\rvert$ | $\lvert\mathbb{Z}_p\rvert + \big(3+(N_x+2)n_S\big)\lvert\mathbb{G}\rvert$ | $\lvert\mathbb{Z}_p\rvert + (4+n_S)\mathbb{G}$ |
| Keygen. Cost | $\lvert\mathbb{G}_T\rvert + (3\log\mathcal{U}+2\ell+7)\lvert\mathbb{G}\rvert$ | $\lvert\mathbb{G}_T\rvert + (2+3\ell)\lvert\mathbb{G}\rvert$ | $\lvert\mathbb{G}_T\rvert + (2+2\ell+2r)\lvert\mathbb{G}\rvert$ |
| Enc. Cost | $(5+n_S+\log\mathcal{U})E$ | $(4+3n_S)E$ | $(7+n_S)E$ |
| Dec. Cost | $(6+3\ell+\log\mathcal{U}(3+\log\mathcal{U}))E$ | $(3+4\ell+(N_x\ell)E$ | $(4+3\ell+3r)E$ |
| CTupdate. Cost | $(1+n_I)E+(3+2n_I+\log\mathcal{U})P$ | $(2+n_I)E+(1+4n_I)P$ | $(2+n_I+2r)E+(3+2n_I)P$ |
| Trace. Cost | $(2\log\mathcal{U})E$ | $\big(3+n_I(1+N_x)\big)E$ | $(3+3r)E$ |
| | $2E+(4+2n_s)P$ | $4E+(4+2n_S)P$ | $3E+(4+n_S)P$ |

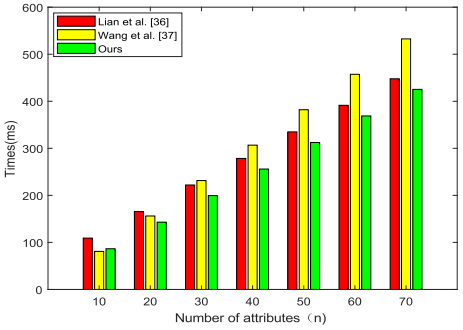


**FIGURE 3.** Computational time of encryption algorithm ($\mathcal{U}=2^4, R=2^2$).

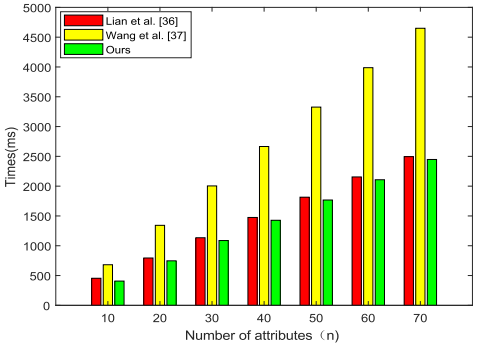


**FIGURE 4.** Computational time of decryption algorithm ($\mathcal{U}=2^4, r=2^2$).

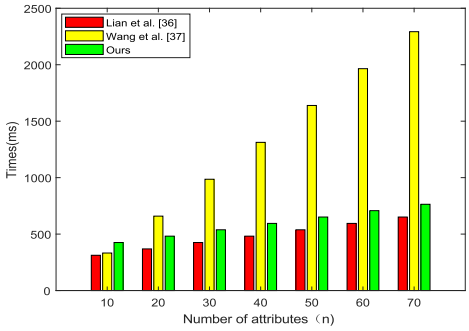


**FIGURE 5.** Computational time of encryption algorithm ($\mathcal{U}=2^{10}, r=2^6$).

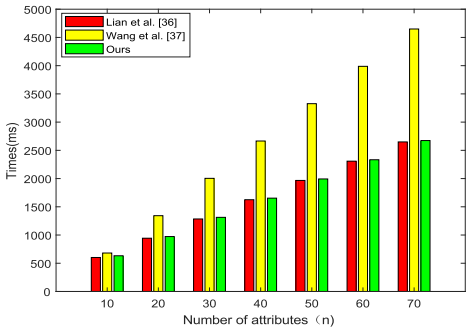


**FIGURE 6.** Computational time of decryption algorithm ($\mathcal{U}=2^{10}, r=2^6$).

and [37]. Since we use the revocation list as a technique to revoke the malicious users, it leads to an increase in ciphertext size. But compared with [36] and [37], we can achieve instant revocation and ensure the forward security, which has great significance in practical application. Meanwhile, the storage cost of secret keys in the user end is lower than others. And for a more realistic performance analysis, we test the efficiency of our scheme and [36], [37] based on Pairing-based Cryptography (PBC) Library. The experiment is executed by a laptop computer configured as Genuine, Intel, CPU, TI500@3.40Ghz, and 2GB RAM. Moreover, we implement our scheme on Type A supersingular elliptic curve $E(F_p)$ : $y^2 = x^3 + x$ with the embedding degree 2, and tested order $p$ is 160 bits.

In a cryptosystem, the computing resources of the user end are often limited, and thus system performance can be effectively evaluated by the computational overhead of the user end. Furthermore, since encryption or decryption time is one of most concerned indicators, we compared the computational time of encryption and decryption phase under different number of the revoked users and attribute universe. In Fig. 3 and Fig.4, we consider a lightweight system with $U = 2^4 = 16$, and $R = 2^2 = 4$. Then we consider a large system with $U = 2^{10} = 1024$, and $R = 2^6 = 64$ in the attribute universe from 10 to 70 in Fig. 5 and Fig. 6

As can be seen from Fig.3 to Fig.6, although the time consumed of encryption and decryption algorithms in our scheme is linear with the number of the revoked users and the number of attributes, we can achieve the same system efficiency as [36]. And it is obvious that the proposed scheme is much more efficient than Wang *et al.*'scheme [37]. Especially in the decryption algorithm, since a bilinear pair operation takes longer time than an exponential, and a fewer bilinear pair operations are performed in the proposed scheme than [37], which makes our scheme more efficient.

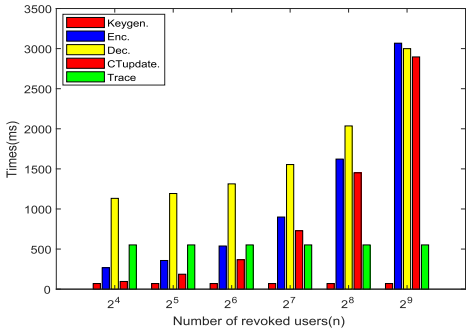


**FIGURE 7.** Computational time of each algorithm ($U = 30, \mathcal{U} = 2^{10}$).

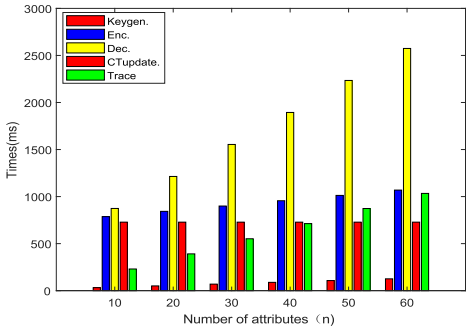


**FIGURE 8.** Computational time of each algorithm ($\mathcal{U} = 2^{10}, r = 2^7$).

Fig.7 and Fig. 8 display the simulated data of the running time of each stage of the proposed scheme. In Fig.7, we set the size of attribute universe $U = 30$, and evaluate the efficiency of revocation mechanism in our system, where the number of revoked users is grow from $2^4 = 16$ to $2^9 = 512$. In Fig.8, the number of revoked users is $2^7 = 128$, and the number of system users is $\mathcal{U} = 2^{10} = 1024$. As shown that the simulated results is consistent with the theoretical analysis, and thus our system is effective.

## VI. CONCLUSIONS

In this paper, we have proposed an updatable CP-ABE scheme, which can support white-box traceability and instant traitor revocation. In our construction, given a secret key, the malicious original key owner will be traced and validly revoked from the cryptosystem. Moreover, we completed the traceability proof of the proposed scheme based on the $l$-SDH assumption, and reduced the IND-CPA security to the decisional $q$-BDHE assumption under standard model.

However, our scheme just achieve the white-box traceability, which is not a strong traceability model. A more realistic situation is that the malicious users will leak the decryption black-box/devices rather than their secret keys, which we define as black-box model. Specifically, the malicious users can hide the decryption algorithm by adjusting the decryption algorithm and secret keys. In this case, since the secret key and the decryption algorithm are not well-formed, the white-box traceable system will fail. For further study, we will try our best to construct an applicable CP-ABE scheme that can achieve black-box traceability and support efficient traitor revocation and ciphertext updating.

## REFERENCES

[1] A. Sahai and B. Waters, "Fuzzy identity-based encryption," in *Advances in Cryptology—EUROCRYPT* (Lecture Notes in Computer Science), vol. 3494, R. Cramer, Ed. Berlin, Germany: Springer-Verlag, 2005, pp. 457–473.

[2] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in *Proc. IEEE Symp. Secur. Privacy-S&P*, Washington, DC, USA, May 2007, pp. 321–334.

[3] V. Goyal, O. Pandey, and A. Sahai, "Attribute-based encryption for fine-grained access control of encrypted data," in *Proc. 13th ACM Conf. Comput. Commun. Secur. (CCS)*, A. Juels, R. N. Wright, and S. D. Vimercati, Eds. New York, NY, USA: ACM, Oct./Nov. 2006, pp. 89–98.

[4] R. Ostrovsky, A. Sahai, and B. Waters, "Attribute-based encryption with non-monotonic access structures," in *Proc. 14th ACM Conf. Comput. Commun. Secur. (CCS)*, P. Ning, S. D. C. di Vimercati and P. F. Syverson, Eds. New York, NY, USA, Oct./Nov. 2007, pp. 195–203.

[5] B. Waters, "Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization," in *Public Key Cryptography—PKC* (Lecture Notes in Computer Science), vol. 6571. D. Catalano, N. Fazio, R. Gennaro, and A. Nicolosi, Eds. Berlin, Germany: Springer, 2011, pp. 53–70.

[6] A. Lewko, T. Okamoto, A. Sahai, K. Takashima, and B. Waters, "Fully secure functional encryption: Attribute-based encryption and (hierarchical) inner product encryption," in *Advances in Cryptology—EUROCRYPT* (Lecture Notes in Computer Science), vol. 6110, H. Gilbert, Ed. Berlin, Germany: Springer, 2010, pp. 62–91.

[7] S. Hohenberger and B. Waters, "Attribute-based encryption with fast decryption," in *Public-Key Cryptography—PKC* (Lecture Notes in Computer Science), vol. 7778, K. Kurosawa and G. Hanaoka, Eds. Berlin, Germany: Springer, 2013, pp. 162–179.

[8] J. Li, W. Yao, Y. Zhang, H. Qian, and J. Han, "Flexible and fine-grained attribute-based data storage in cloud computing," *IEEE Trans. Services Comput.*, vol. 10, no. 5, pp. 785–796, Jan. 2016.

[9] A. Lewko and B. Waters, "New proof methods for attribute-based encryption: Achieving full security through selective techniques," in *Advances in Cryptology—CRYPTO* (Lecture Notes in Computer Science), vol. 7417, R. Safavi-Naini and R. Canetti, Eds. Berlin, Germany: Springer, Aug. 2012, pp. 180–198.

[10] M. Chase and S. S. Chow, "Improving privacy and security in multi-authority attribute-based encryption," in *Proc. 16th ACM Conf. Comput. Commun. Secur. (CCS)*, E. Al-Shaer, S. Jha, A. D. Keromytis, Eds. New York, NY, USA: ACM, Nov. 2009, pp. 121–130.

[11] N. Agrawal and S. Tapaswi, "A trustworthy agent-based encrypted access control method for mobile cloud computing environment," *Pervasive Mobile Comput.*, vol. 52, pp. 13–28, Jan. 2019.

[12] V. K. A. Sandor, Y. Lin, X. Li, F. Lin, and S. Zhang, "Efficient decentralized multi-authority attribute based encryption for mobile cloud data storage," *J. Netw. Comput. Appl.*, vol. 129, pp. 25–36, Mar. 2019.

[13] M. J. Hinek, S. Jiang, R. Safavi-Naini, and S. F. Shahandashti, "Attribute-based encryption without key cloning" *Int. J. Appl. Cryptol.*, vol. 2, no. 3, pp. 250–270, Feb. 2012.

[14] J. Li, K. Ren, and K. Kim, "A2BE: Accountable attribute-based encryption for abuse free access control," *IACR Cryptol. ePrint Arch.*, vol. 2009, p. 118, 2009.

[15] S. Yu, K. Ren, W. Lou, and J. Li, "Defending against key abuse attacks in KP-ABE enabled broadcast systems," in *Security and Privacy in Communication Networks. SecureComm*, Y. Chen, T. D. Dimitriou, and J. Zhou, Eds., vol. 19. Berlin, Germany: Springer, Sep. 2009, pp. 311–329.

[16] Z. Liu, Z. Cao, and D. S. Wong, "White-box traceable ciphertext-policy attribute-based encryption supporting any monotone access structures," *IEEE Trans. Inf. Forensics Security*, vol. 8, no. 1, pp. 76–88, Jan. 2013.

[17] J. Ning, Z. Cao, X. Dong, L. Wei, and X. Lin, "Large universe ciphertext-policy attribute-based encryption with white-box traceability," in *Proc. Eur. Symp. Res. Comput. Secur. (ESORICS)* (Lecture Notes in Computer Science), vol. 8713, M. Kutyłowski, J. Vaidya, Eds. Cham, Switzerland: Springer, 2014, pp. 55–72.

[18] J. Ning, X. Dong, Z. Cao, L. Wei, and X. Lin, "White-box traceable ciphertext-policy attribute-based encryption supporting flexible attributes," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 6, pp. 1274–1288, Jun. 2015.

[19] Y. Jiang, W. Susilo, Y. Mu, and F. Guo, "Ciphertext-policy attribute-based encryption against key-delegation abuse in fog computing," *Future Gener. Comput. Syst.*, vol. 78, pp. 720–729, Jan. 2018.
[20] G. Yu, Y. Wang, Z. Cao, J. Lin, and X. Wang, "Traceable and undeniable ciphertext-policy attribute-based encryption for cloud storage service," *Int. J. Distr. Sensor Netw.*, vol. 15, no. 4, p. 1550147719841276, 2019.
[21] A. Boldyreva, V. Goyal, and V. Kumar, "Identity-based encryption with efficient revocation," in *Proc. 15th ACM Conf. Comput. Commun. Secur. (CCS)*, New York, NY, USA, Oct. 2008, pp. 417–426.
[22] J. H. Seo and K. Emura, "Revocable identity-based encryption revisited: Security model and construction," in *Public-Key Cryptography—PKC* (Lecture Notes in Computer Science), vol. 7778, K. Kurosawa and G. Hanaoka, Eds. Berlin, Germany: Springer, 2013, pp. 216–234.
[23] A. Takayasu and Y. Watanabe, "Lattice-based revocable identity-based encryption with bounded decryption key exposure ristance," in *Proc. Australas. Conf. Inf. Secur. Privacy-ACISP* (Lecture Notes in Computer Science), vol. 10342, J. Pieprzyk and S. Suriadi, Eds. Cham, Switzerland: Springer, 2017, pp. 184–204.
[24] A. Lewko, A. Sahai, and B. Waters, "Revocation systems with very small private keys," in *Proc. IEEE Symp. Secur. Privacy-S&P*, May 2010, pp. 273–285.
[25] A. Sahai, H. Seyalioglu, and B. Waters, "Dynamic credentials and ciphertext delegation for attribute-based encryption," in *Advances in Cryptology—CRYPTO* (Lecture Notes in Computer Science), vol. 7417. Berlin, Germany: Springer, 2012, pp. 199–217.
[26] 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, Jul. 2011.
[27] J. Li, W. Yao, J. Han, Y. Zhang, and J. Shen, "User collusion avoidance CP-ABE with efficient attribute revocation for cloud storage," *IEEE Syst. J.*, vol. 12, no. 2, pp. 1767–1777, Jun. 2018.
[28] 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* (Lecture Notes in Computer Science), vol. 10892, B. Preneel and F. Vercauteren, Eds. Cham, Switzerland: Springer, Jul. 2018, pp. 516–534.
[29] Y. Luo, M. Xu, K. Huang, D. Wang, and S. Fu, "Efficient auditing for shared data in the cloud with secure user revocation and computations outsourcing," *Comput. Secur.*, vol. 73, pp. 492–506, Mar. 2018.
[30] S. Xu, G. Yang, and Y. Mu, "Revocable attribute-based encryption with decryption key exposure resistance and ciphertext delegation," *Inf. Sci.*, vol. 479, pp. 116–134, Apr. 2019.
[31] H. Ma, Z. Wang, and Z. Guan, "Efficient ciphertext-policy attribute-based online/offline encryption with user revocation," *Secur. Commun. Netw.*, vol. 2019, Feb. 2019, Art. no. 8093578.
[32] K. Lee, S. G. Choi, D. H. Lee, J. H. Park, and M. Yung, "Self-updatable encryption: Time constrained access control with hidden attributes and better efficiency," in *Advances in Cryptology—ASIACRYPT* (Lecture Notes in Computer Science), vol. 8269, K. Sako and P. Sarkar, Eds. Berlin, Germany: Springer, 2013, pp. 235–254.
[33] K. Lee, "Self-updatable encryption with short public parameters and its extensions," *Des., Codes Cryptograph.*, vol. 79, no. 1, pp. 121–161, 2016.
[34] Z. Liu and D. S. Wong, "Practical attribute-based encryption: Traitor tracing, revocation and large universe," in *Proc. Int. Conf. Appl. Cryptograph. Netw. Secur.* (Lecture Notes in Computer Science), vol. 9092, T. Malkin, V. Kolesnikov, A. Lewko, M. Polychronakis, Eds. Cham, Switzerland: Springer, Jun. 2015, pp. 127–146.
[35] J. Ning, Z. Cao, X. Dong, and L. Wei, "Traceable and revocable CP-ABE with shorter ciphertexts," *Sci. China Inf. Sci.*, vol. 59, p. 119102, Nov. 2016. Accessed: Sep. 5, 2016. doi: 10.1007/s11432-016-0062-7.
[36] H. Lian, G. Wang, and Q. Wang, "Fully secure traceable and revocable-storage attribute-based encryption with short update keys via subset difference method," in *Proc. 3rd Int. Conf. Secur. Smart Cities, Ind. Control Syst. Commun. (SSIC)*, Shanghai, China, Oct. 2018, pp. 1–8.
[37] S. Wang, K. Guo, and Y. Zhang, "Traceable ciphertext-policy attribute-based encryption scheme with attribute level user revocation for cloud storage," *PLoS ONE*, vol. 13, no. 10, p. e0206952, 2018. Accessed: Sep. 13, 2018. doi: 10.1371/journal.pone.0203225.
[38] Z. Liu, S. Duan, P. Zhou, and B. Wang, "Traceable-then-revocable ciphertext-policy attribute-based encryption scheme," *Future Gener. Comput. Syst.*, vol. 93, pp. 903–913, Oct. 2019.
[39] D. Boneh and M. Franklin, "Identity-based encryption from the Weil pairing," in *Advances in Cryptology—CRYPTO*, vol. 2139. Berlin, Germany: Springer, 2001, pp. 213–229.
[40] A. Beimel, "Secure schemes for secret sharing and key distribution," Ph.D. dissertation, Dept. Comput. Sci., Israel Inst. Technol., Technion, Haifa, Israel, 1996.
[41] D. Boneh and X. Boyen, "Short signatures without random oracles," in *Advances in Cryptology—EUROCRYPT*, (Lecture Notes in Computer Science), vol. 3027. Berlin, Germany: Springer-Verlag, 2004, pp. 56–73.
[42] S. Goldwasser and S. Micali, "Probabilistic encryption," *J. Comput. Syst. Sci.*, vol. 28, no. 2, pp. 270–299, Apr. 1984.

**ZHENHUA LIU** received the B.S. degree from Henan Normal University, in 2000, and the master's and Ph.D. degrees from Xidian University, China, in 2003 and 2009, respectively, where he is currently a Professor. His research interests include cryptography and information security.

**JING XU** received the B.S. degree from Henan Normal University, in 2017. She is currently pursuing the master's degree in applied mathematics with Xidian University, China. Her research interests include cryptography and cloud security.

**YAN LIU** received the B.S. degree from Shenyang Agricultural University, in 2017. She is currently pursuing the master's degree in applied mathematics with Xidian University, China. Her research interests include cryptography and cloud security.

**BAOCANG WANG** received the B.S. degree in computational mathematics and the M.S. and Ph.D. degrees in cryptology from Xidian University, China, in 2001, 2004, and 2006, respectively, where he is currently a Professor and a Ph.D. Supervisor. His research interests include post-quantum cryptography, fully homomorphic cryptography, number theoretic algorithms, and cloud security.