IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received January 29, 2019, accepted February 18, 2019, date of publication March 13, 2019, date of current version March 26, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2902040

Hidden Ciphertext Policy Attribute-Based Encryption With Fast Decryption for Personal Health Record System

LEYOU ZHANG[1], GONGCHENG HU[1], YI MU[2,3], (Senior Member, IEEE), AND FATEMEH REZAEIBAGHA[4]

[1]School of Mathematics and Statistics, Xidian University, Xi'an 710126, China
[2]Fujian Provincial Key Laboratory of Network Security and Cryptology, College of Mathematics and Informatics, Fujian Normal University, Fuzhou 350007, China
[3]School of Computer Science and Software Engineering, University of Wollongong, Wollongong, NSW 2522, Australia
[4]SMART Infrastructure, University of Wollongong, Wollongong, NSW 2522, Australia

Corresponding authors: Gongcheng Hu (gchenghu@126.com) and Yi Mu (ymu.ieee@gmail.com)

This work was supported in part by the National Nature Science Foundation of China under Grant 61872087, in part by the National Cryptography Development Fund under Grant MMJJ20180209, in part by the Key Research Project of Shaanxi Province under Grant 2018GY-018, and in part by the Foundation of Education Department of Shaanxi Province under Grant 17JK0713.

ABSTRACT Since cloud computing has been playing an increasingly important role in real life, the privacy protection in many fields has been paid more and more attention, especially, in the field of personal health record (PHR). The traditional ciphertext-policy attribute-based encryption (CP-ABE) provides the fine-grained access control policy for encrypted PHR data, but the access policy is also sent along with ciphertext explicitly. However, the access policy will reveal the users' privacy, because it contains too much sensitive information of the legitimate data users. Hence, it is important to protect users' privacy by hiding access policies. In most of the previous schemes, although the access policy is hidden, they face two practical problems: 1) these schemes do not support large attribute universe, so their practicality in PHR is greatly limited and 2) the cost of decryption is especially high since the access policy is embedded in the ciphertext. To address these problems, we construct a CP-ABE scheme with efficient decryption, where both the size of public parameters and the cost of decryption are constant. Moreover, we also show that the proposed scheme achieves full security in the standard model under static assumptions by using the dual system encryption method.

INDEX TERMS Personal health record (PHR), attribute-based encryption, hidden policy, fast decryption.

I. INTRODUCTION

As an emerging technology in recent years, cloud computing provides a fast and efficient way to share data resources, and a mountain number of people access data through the network. For example, in the personal system health record system, a patient does not have to carry various paper versions of the test forms to make a diagnosis according to the traditional way, but he/she can store, retrieve and share the health record only by uploading his own personal health record to the PHR system. A patient has the full control to his/her own PHR document and authorizes who can access these health data, such as friends, family or healthcare providers. In order to achieve accurate access control of PHR, data owners urgently need a kind of encryption scheme that can realize fine-grained access control [22], [28], [30], [32].

Hidden ciphertext policy attribute-based encryption scheme provides a good way to solve the problem, where it achieves privacy protection by hiding access control policy. However, In the previous mechanisms [2], [3], [7], the access control policy is often sent along with ciphertext explicitly, which makes it easy reveal the users' privacy, since some attributes in access structure carry crucial identity information of the legitimate users. In PHR, an access policy defined by a patient may contain some sensitive attributes such as cardiologist, central hospital and so on [8], [31]. Therefore, for an unauthorized user, even if he cannot decrypt successfully, he can also infer from the access policy in

The associate editor coordinating the review of this manuscript and approving it for publication was Constantinos Marios Angelopoulos.

cleartext form which the encryptor suffers from some disease. The first hidden ciphertext-policy attribute-based encryption (HCP-ABE) was introduced in [16], where the access structure was embedded in the ciphertext and not sent directly. Subsequently, some other hidden CP-ABE schemes were also successively proposed in [17]–[19]. However, access structures in these schemes only support AND gates or AND gates on positive, negative and wildcard. These lead to two drawbacks. First, the size of public parameters increases linearly with the number of attributes, and secondly, the cost of the decryption is greatly increased. Due to the above drawbacks, some low-overhead schemes are introduced in [13] and [14] and the common method adopted by these schemes is to introduce a decryption test by adding some redundant components to a ciphertext before the decryption stage. Although the above schemes improve the efficiency of decryption, the length of ciphertext is also significantly increased and this will become a bottleneck restricting higher performance. Additionally, these schemes are extremely vulnerable to decisional Diffie-Hellman test (DDH-test) attack ([9], [20], [27]–[29]).

A. RELATED WORK

Since Attribute-Based Encryption was first proposed by Sahai and Waters [6], it has been seen as the most promising approach for fine-grained access control in the field of cloud computing. With the continuous improvements of ABE, currently, there are mainly two basic types of ABE schemes, Key Policy ABE (KP-ABE) [24], [26] and Ciphertext Policy ABE (CP-ABE) [7], [10], [13]. In KP-ABE scheme, keys are associated with access structure and ciphertexts are associated with a set of attributes. The first KP-ABE scheme was proposed by Wang and He [24]. But in this scheme, the trusted authority fully determines the combination of attributes associated with the ciphertext, because the access control associated with the key are generated by the center for each legitimate decryption user. Then Sahai et al. proposed another KP-ABE scheme, in which the decryption keys of users' could express any access formulas over attributes, including non-monotone ones [25].

The first CP-ABE scheme was introduced in [7], where ciphertexts were associated with access structure defined by data owners and the key are associated with sets of attributes about users. Subsequently, there are a lot of CP-ABE schemes were also successively proposed in [15], [17], [18], and [21], but these schemes only support AND gates. To realize the access structure more expressive, Waters proposed an access structure based a linear secret sharing scheme (LSSS), and it is also a provably secure scheme under the standard model [3]. In order to further protect users' privacy, the first the CP-ABE scheme with hidden access structure was proposed by Nishide et al. [16]. In their work, access control policy isn't sent along with ciphertext explicitly, in other words, no unauthorized user can obtain useful information about the access structure. Some other schemes with the same performance have been proposed by other researchers, which are called Anonymous Attribute-Based Encryption [22], [29]. In these schemes, only sets of the user satisfying the access policy was embedded in the ciphertext, then the user can successfully decrypt the ciphertext. Later, authors in introduced another highly effective anonymous CP-ABE scheme, and its security proof was given under the Decisional Modified Bilinear Diffie-Hellman assumption (MBDH) [20]. However, their work only gives a general analysis and lacks detailed security proof. Some other works were proposed in [9], [14], and [27] to make further improvements on anonymous CP-ABE scheme. Unfortunately, all of them have to face high-overhead of decryption, which may make them lose their practicability.

B. OUR CONTRIBUTION

In recent years, with the rapid development of internet and cloud computing, a mountain number of Intelligent Medical Systems have been designed. However, in the previous mechanisms based attribute encryption, access control policies are often sent along with ciphertext, which makes it easy to reveal the sensitive information of users in the system. Especially, in PHR, the specific attribute values in a access policy carry much more sensitive information, such as the patient's pulse frequency, his family history of hereditary diseases, the result of the patient's laboratory test report and so on. In order to deal with the above problems, our contributions mainly include the following three parts.

- **Access structure**: Each attribute in this paper contains two parts, attribute name index and its attribute value. And Each attribute has multiple candidate values. Every decrytor only knows the attribute name index of his own and his attribute value. Moreover, the values of the attributes in the access policy defined by the encryptor are hidden, and they are not sent with the ciphertext. Only the access matrix and the defined function ρ are sent to the decryptor along with the ciphertext. What's more, the proposed scheme can handle any access control policy that can be expressed as a linear secret sharing scheme.
- **Fast decryption**: Obviously, it is hard for a user to know whether his attribute set satisfies the access policy defined by the encryptor, if the access policy associated with a ciphertext is fully hidden. Therefore, a decryptor has to do a lot of calculations to determine whether he is legal or not. In this paper, we present an efficient construction of Hidden Ciphertext Policy Attribute-Based Encryption Supports Fast Decryption, where, the number of bilinear pairing evaluations is reduced to a constant in decryption phase.
- **Data verifiability**: In most previous schemes, there are usually two practical problems deserve to be considered. one is the size of the public parameters increases linearly with the size of the universe. And the other is the authorized user cannot determine whether the message he obtained through decryption is valid or not, because there is no verifiable link to the message. However, in the proposed scheme, the size of public parameters is constant, so the attribute universe in this scheme can be exponentially large and it also supports

validation of decrypted messages, which can further improve the reliability of decryption. Furthermore, we also prove the full security of the proposed scheme in the standard model under static assumptions by using the dual system encryption method [1].

C. ORGANIZATION

The remainder of the paper is structured as follows. In section 2, some preliminary concepts are introduced, such as composite-order bilinear map, access structure and complexity assumptions. We describe the definition of our proposed algorithm and its security model in the section 3. The specific structure of the proposed scheme is presented in section 4. Section 5 is a detailed description of the full security proof. Finally, we give a brief conclusion and performance analysis about the proposed scheme.

II. PRELIMINARIES

A. COMPOSITE ORDER BILINEAR GROUPS

Our scheme is based on composite order bilinear group whose order is the product of four distinct primes. Let $\mathscr{G}$ be an algorithm which takes a security parameter 1^λ as input and output a tuple $(\mathbb{G}, \mathbb{G}_T, e, p_1, p_2, p_3, p_4)$. Where p_1, p_2, p_3, p_4 are distinct primes, $\mathbb{G}$ and $\mathbb{G}_T$ are cyclic groups of order $N = p_1p_2p_3p_4$, and $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ is a map such that

1. Bilinear: $\forall g, w \in \mathbb{G}, a, b \in \mathbb{Z}_N, e(g^a, w^b) = e(g, w)^{ab}$.

2. Non-degengerate: $\exists g \in \mathbb{G}$ such that $e(g, g)$ has order N in $\mathbb{G}_T$. We further require that multiplication in $\mathbb{G}$, $\mathbb{G}_T$ and the bilinear map e,are computable in polynomial time in λ. Let $\mathbb{G}_{P_1}, \mathbb{G}_{P_2}, \mathbb{G}_{P_3}, \mathbb{G}_{P_4}$ denote that subgroups of $\mathbb{G}$, respectively. Note also that if $g_1 \in \mathbb{G}_{P_1}, g_2 \in \mathbb{G}_{P_2}$ then $e(g_1, g_2) = 1$. In fact, $g_{p_j}, (j = 1, 2, 3, 4)$ be the generator of $\mathbb{G}_{P_j}$, respectively. Hence, $\forall \alpha_j \in \mathbb{Z}_N$, then $e(g_{p_j}^{\alpha_j}, g_k^{\alpha_k}) = 1, (j \neq k)(j, k = 1, 2, 3, 4)$.

B. ACCESS STRUCTURES

Definition 1 (Access Struture [2]). Let $\Omega = \{P_1, P_2, \ldots, P_n\}$ be a set of parties. A collection $\mathbb{A} \subseteq 2^\Omega$ is monotone if $\forall D, F$: if $B \in \mathbb{A}$ and $D \subseteq F$ then $F \in \mathbb{A}$. An access structure (respectively, monotone access structure)is a collection (respectively, monotone collection)$\mathbb{A}$ of non-empty subsets of $\{P_1, P_2, \ldots, P_n\}$, *i.e.*, $\mathbb{A} \subseteq 2^\Omega \setminus \{0\}$. The sets in $\mathbb{A}$ are called the authorized sets, and the sets not in $\mathbb{A}$ are called the authorized sets.

C. LINEAR SECRET SHARING SCHEMES

Let U be the attribute universe, which has n categories of attributes, and $U = (ATT_1, ATT_2, ATT_3, \ldots, ATT_n)$. Each attribute $ATT_x \in U$ has n_x values, $VU_x = \{\xi_{x,1}, \xi_{x,2}, \xi_{x,3}, \ldots, \xi_{x,n_x}\}$ is the set of all possible values for the x^{th} attribute ATT_x in U. In our construction, each attribute includes two parts: attribute name and its value. Where A is an $l \times n$ matrix over $\mathbb{Z}_p$, which called the share-generating matrix. ρ is a map from each row of A to an attribute name index. (*i.e*, $\rho : \{1, 2, 3, \ldots, l\} \to \{1, 2, 3, \ldots, n\}$). A secret value s can be shared as follows.

Let s be a secret value to be shared, considering a column vector $V = (s, y_2, y_3, \ldots, y_n)$ $(s \in \mathbb{Z}_p, \{y_2, y_3, \ldots, y_n\} \in \mathbb{Z}_p)$, then $\lambda_x = A_x \times V(A_x$ denotes the x^{th} row of $A)$, where λ_x is a share of s $(x = 1, 2, 3, \ldots, l)$. If P is any authorized attribute name index set, and $I = \{x|\rho(x) \in P\} \subseteq \{1, 2, 3, \ldots, l\}$, then there exist constants $\{\omega_x\}_{x\in I}$ such that $\sum_{x\in I} \omega_x A_x = (1, 0, 0, \ldots, 0)$, and the secret value s can be reconstruction by $\sum_{x\in I} \omega_x \lambda_x$. Namely, if S' is an unauthorized set, and $I' = \{x \mid \rho_{(x)} \in I'_L\}$, there exist $\{\omega'_x\}_{x\in I'}$ and the first component ω'_1 is any non-zero element, then $\sum_{x\in I'} \omega'_x \lambda_x = 0$. A subset I of $\{1, 2, 3, \ldots, l\}$ is said to be a minimum authorized attribute name index set of the access policy, if I satisfies (A, ρ) and its any subset I_{sub} does not satisfy (A, ρ), we define the minimum authorized attribute name index sets of (A, ρ) as I_{min}. We express our access policy by $(A, \rho, \mathcal{T})$, where, $\mathcal{T} = (t_{\rho_{(1)}}, t_{\rho(2)}, t_{\rho(3)}, \ldots, t_{\rho(l)})(t_{\rho_{(x)}} \in VU_{\rho_{(x)}})$ is the attribute value set associated with the access policy (A, ρ). Suppose a user has an attribute set $S = (I_S, L_S)$, the user satisfying the access policy, if there exists an $I = \{x|\rho(x) \in P\} \subseteq I_S$ and $H(L_{\rho_{(x)}}) = H(t_{\rho_{(x)}})$. I_S denotes the user's attribute name index set and $L_S = \{L_1, L_2, \ldots, L_l\}(L_i \in VU_i)$ is its attribute value set.

Example: We assume that there are four categories of attributes in a PHR system as shown in the Fig.1 and set attribute universal $U = (i.e$, *Hospital, Department, Sex, Doctor number*). Without loss of generality, considering their attribute name index set is $\{num_1, num_2, num_3, num_4\}_{num_i\in\{1,2,\ldots,n\}}$. The attribute 'Hospital' has four values $(\xi_{1,1}, \xi_{1,2}, \xi_{1,3}, \xi_{1,4})$, 'Department' has four values $(\xi_{2,1}, \xi_{2,2}, \xi_{2,3}, \xi_{2,4})$, 'Sex' has two values $(\xi_{3,1}, \xi_{3,2})$ and the attribute 'Doctor number' has four values $(\xi_{4,1}, \xi_{4,2}, \xi_{4,3}, \xi_{4,4})$. Suppose there is a patient in PHR who allows all male doctors in the genetics department of city hospital to view his personal health record, and he can definite the access policy $(A, \rho, \mathcal{T})$ as shown in the Fig.2, where $\mathcal{T} = (t_{\rho_{(k_1)}}, t_{\rho_{(k_2)}}, t_{\rho_{(k_3)}}) = (Cityhospital, Genetics, Male)$. Then, his personal health records are encrypted by the access policy $(A, \rho, \mathcal{T})$ and uploaded them to the cloud. If a user with attribute set $S = (I_S, L_S)$ wants to decrypt the document, he first computes the values of k_1, k_2, k_3 $(i.e, k_1, k_2, k_3 \in \{1, 2, 3, \ldots, l\})$ by his attribute name index I_S and the map ρ associated with (A, ρ). Obviously, a data user is legal if and only if his attribute values are $(Cityhospital, Genetics, Male)$ that is his attribute name index $I_S = (num_1, num_2, num_3)$. Then, the user calculates the value of ω_j by the equation $\sum_{j=k_1}^{k_3} M_j\omega_j = (1, 0, 0, \ldots, 0)$ and calculates the secret value s by the equation $\Sigma_{j=k_1}^{k_3} \lambda_j\omega_j$. If the user's attribute values set $L_S = (Cityhospital, Genetics, Feale)$, it is obvious that $H(t_{\rho_{(k_3)}}) \neq H(L_{\rho_{(k_3)}})$ and the user failed to decrypt the document. We also point that a data user with attribute set S satisfying the access policy associated with one ciphertext if and only if $H(t_{\rho_{(x)}}) = H(L_{\rho_{(x)}})$, for $\{x|\rho(x) \in I_S\}$.

D. COMPLEXITY ASSUMPTION

We now state the complexity assumptions used in this paper. The first three assumptions are the same assumptions as

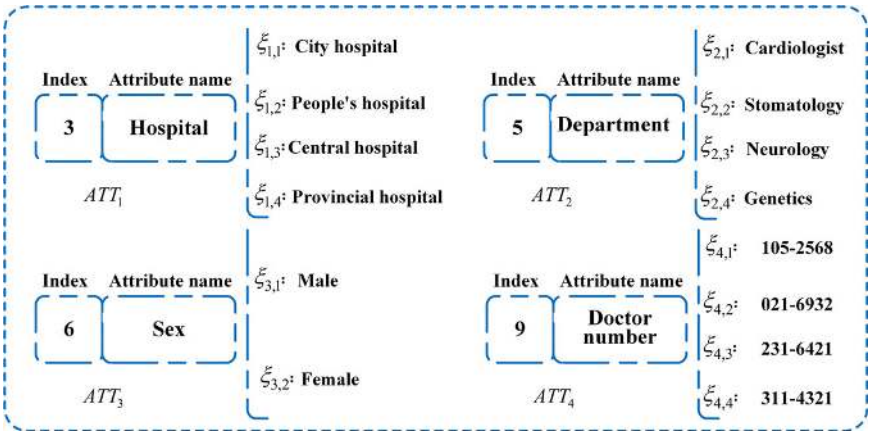

FIGURE 1. Examples of attribute categories in PHR.

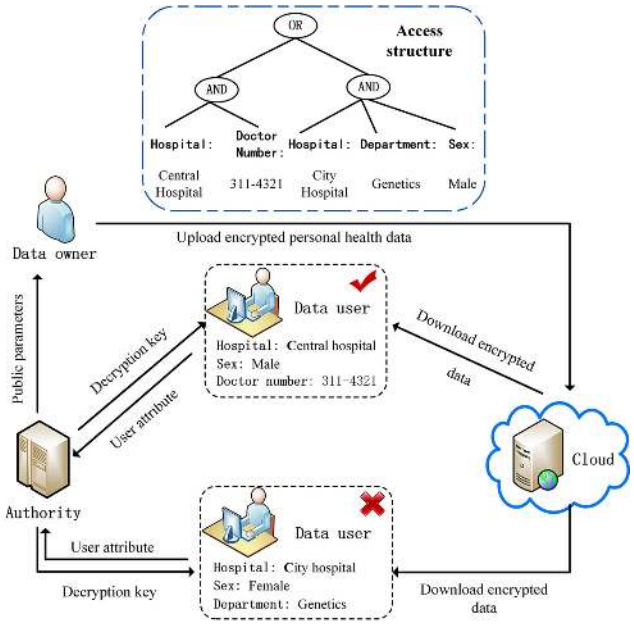

FIGURE 2. Example of PHR cloud storage.

in [5], but the group in our scheme whose order is a product of four primes which are different from [23]. *Assumption 4* was used in [8].

Assumption 1. Let $\mathscr{G}$ be the algorithm mentioned above and define the following distribution:

$$\begin{aligned}(p_1, p_2, p_3, p_4, \mathbb{G}, \mathbb{G}_T, e) &\leftarrow \mathscr{G}(1^\lambda), N = p_1p_2p_3p_4\\ g &\xleftarrow{R} \mathbb{G}_{p_1}, X_3 \xleftarrow{R} \mathbb{G}_{p_3}, X_4 \xleftarrow{R} \mathbb{G}_{p_4}\\ D &= (\mathbb{G}, \mathbb{G}_T, N, e, g, X_3, X_4)\\ T_1 &\xleftarrow{R} \mathbb{G}_{p_1} \times \mathbb{G}_{P_2}, T_2 \xleftarrow{R} \mathbb{G}_{p_1}\end{aligned}$$

The advantage of $\mathscr{A}$ in breaking this assumption is defined as

$$Adv^1_{\mathscr{A}} = |\Pr[\mathscr{A}(D, T_1) = 1] - \Pr[\mathscr{A}(D, T_2) = 1]|$$

Definition 2: If the algorithm $\mathscr{G}$ satisfies assumption 1, for any polynomial time adversary $\mathscr{A}$, its advantage $Adv^1_{\mathscr{A}}$ is negligible.

Assumption 2. Let $\mathscr{G}$ be the algorithm mentioned above and define the following distribution:

$$\begin{aligned}(p_1, p_2, p_3, p_4, \mathbb{G}, \mathbb{G}_T, e)&\\ \leftarrow \mathscr{G}(1^\lambda), N = p_1p_2p_3p_4&\\ (g, X_1 \xleftarrow{R} \mathbb{G}_{p_1}), (X_2, Y_2 \xleftarrow{R} \mathbb{G}_{p_2}),&\\ (X_3, Y_3 \xleftarrow{R} \mathbb{G}_{p_3}), X_4 \xleftarrow{R} \mathbb{G}_{p_4}&\\ D = (\mathbb{G}, \mathbb{G}_T, N, e, g, X_1X_2, Y_2Y_3, X_3, X_4)&\\ T_1 \xleftarrow{R} \mathbb{G}_{p_1} \times \mathbb{G}_{P_2} \times \mathbb{G}_{P_3}, T_2 \xleftarrow{R} \mathbb{G}_{p_1} \times \mathbb{G}_{p_3}&\end{aligned}$$

The advantage of $\mathscr{A}$ in breaking this assumption is defined as

$$Adv^2_{\mathscr{A}} = |\Pr[\mathscr{A}(D, T_1) = 1] - \Pr[\mathscr{A}(D, T_2) = 1]|$$

Definition 3: If the algorithm $\mathscr{G}$ satisfies assumption 2, for any polynomial time adversary $\mathscr{A}$, its advantage $Adv^2_{\mathscr{A}}$ is negligible.

Assumption 3. Let $\mathscr{G}$ be the algorithm mentioned above and define the following distribution:

$$\begin{aligned}(p_1, p_2, p_3, p_4, \mathbb{G}, \mathbb{G}_T, e)&\\ \leftarrow \mathscr{G}(1^\lambda), N = p_1p_2p_3p_4&\\ \alpha, s \in \mathbb{Z}_N, g \xleftarrow{R} \mathbb{G}_{P_1}, (g_2, X_2, Y_2 \xleftarrow{R} \mathbb{G}_{p_2}),&\\ X_3 \xleftarrow{R} \mathbb{G}_{p_3}, X_4 \xleftarrow{R} \mathbb{G}_{p_4}&\\ D = (\mathbb{G}, \mathbb{G}_T, N, e, g, g_2, g^\alpha X_2, g^s Y_2, X_3, X_4)&\\ T_1 = e(g, g_1)^{\alpha s}, T_2 \xleftarrow{R} \mathbb{G}_T&\end{aligned}$$

The advantage of $\mathscr{A}$ in breaking this assumption is defined as

$$Adv^3_{\mathscr{A}} = |\Pr[\mathscr{A}(D, T_1) = 1] - \Pr[\mathscr{A}(D, T_2) = 1]|$$

Definition 4: If the algorithm $\mathscr{G}$ satisfies assumption 3, for any polynomial time adversary $\mathscr{A}$, its advantage $Adv^3_{\mathscr{A}}$ is negligible.

Assumption 4. Let $\mathscr{G}$ be the algorithm mentioned above and define the following distribution:

$$
\begin{aligned}
&(p_1, p_2, p_3, p_4, \mathbb{G}, \mathbb{G}_{\mathbb{T}}, e) \\
&\quad \leftarrow \mathscr{G}(1^{\lambda}), N = p_1p_2p_3p_4 \\
&(r', t' \in \mathbb{Z}_N), \\
&\quad g \overset{R}{\leftarrow} \mathbb{G}_{P_1}, (g_2, X_2, A_2, B_2, D_2 \overset{R}{\leftarrow} \mathbb{G}_{p_2}) \\
&\quad X_3 \overset{R}{\leftarrow} \mathbb{G}_{p_3}, (X_4, A_4, D_4, Z \overset{R}{\leftarrow} \mathbb{G}_{p_4}) \\
&\quad D = (\mathbb{G}, \mathbb{G}_{\mathbb{T}}, N, e, g, g_2, g_1^{t'}B_2, X_3, X_4, g^{r'}D_2D_4) \\
&\quad T_1 = g^{r'}A_2A_4, T_2 \overset{R}{\leftarrow} \mathbb{G}_{P_1} \times \mathbb{G}_{P_2} \times \mathbb{G}_{P_4}
\end{aligned}
$$

The advantage of $\mathscr{A}$ in breaking this assumption is defined as

$$Adv^4_{\mathscr{A}} = |\Pr[\mathscr{A}(D, T_1) = 1] - \Pr[\mathscr{A}(D, T_2) = 1]|$$

Definition 5: If the algorithm $\mathscr{G}$ satisfies assumption 4, for any polynomial time adversary $\mathscr{A}$, its advantage $Adv^4_{\mathscr{A}}$ is negligible.

E. HIDDEN CIPHERTEXT POLICY ATTRIBUTE-BASED ENCRYPTION

A hidden CP-ABE scheme consists of the following four algorithms.

Setup $(1^{\lambda}) \rightarrow (PK, MSK)$: It is a randomized algorithm that takes a security parameter λ as input and outputs the public parameters PK and master key MSK.

KeyGen $(PK, MSK, S) \rightarrow SK$: The key generation algorithm takes the public parameters PK, the master key MSK and the user's attributes set S as input. It outputs the user's private key SK associated with S.

Encrypt $(PK, M, (A, \rho, \mathcal{T})) \rightarrow CT$: The encryption algorithm takes the public parameters PK, a plaintext message M, and an access structure $(A, \rho, \mathcal{T})$ as input, and outputs a ciphertext CT, where T is a set of attribute values in the access structure and not sent along with the ciphertext CT.

Decrypt $(PK, SK, CT) \rightarrow M$: It takes the public parameters PK, a secret key SK associated with the attributes set $S = (I_S, L_S)$,and a ciphertext CT encrypted under access structure (A, ρ) as input, and outputs the message M or a special symbol $\perp$ denotes that a user failed to decrypt the ciphertext CT.

III. SECURITY MODEL

In this part, we give the security model for our scheme. This selective security model is described by a security game between an adversary $\mathscr{A}$ and a challenger $\mathscr{B}$. The game proceeds as follows.

Setup: The challenger $\mathscr{B}$ runs this algorithm (1^{λ}) to output the public parameters PK and the master key MSK. Then, the public parameters are sent to the adversary $\mathscr{A}$.

Phase 1: The adversary $\mathscr{A}$ submits sets of attributes $S_1, S_2, \ldots, S_Q$ to the challenger $\mathscr{B}$ for secret key, where, Q is a polynomial bounded number. The challenger generates secret key SK_{S_i} corresponding to the set of attribute S_i by running the algorithm $(PK, MSK, S_i) \rightarrow SK_{S_i}$.

Challenge: The adversary $\mathscr{A}$ submits two challenge messages M_0, M_1 $(|M_0| = |M_1|)$ and two access structures $(A, \rho, \mathcal{T}_0)$, $(A, \rho, \mathcal{T}_1)$ to the challenger $\mathscr{B}$, with the restriction that none of them can be satisfied by any of the queried attribute sets in phase 1. In response, the challenger $\mathscr{B}$ flips a random coin $b \in \{0, 1\}$, sets $CT_{\mathcal{T}_b}$ is the ciphertext of M_b under the access policy $(A, \rho, \mathcal{T}_b)$, and sends the ciphertext $CT_{\mathcal{T}_b}$ to the adversary.

Phase 2: Phase 1 is repeated. But, none of the sets of attributes $S_{Q+1}, S_{Q+2}, \ldots, S_q$ satisfying the access policy corresponding to the challenger.

Guess: The adversary $\mathscr{A}$ outputs a guess bit $b' \in \{0, 1\}$ and wins the game if $b = b'$.

The advantage of the adversary $\mathscr{A}$ in this game is defined as $|\ Pr[b = b'] - \frac{1}{2}\ |$, where the probability is taken over the random bits used by the adversary $\mathscr{A}$ and the challenger $\mathscr{B}$.

IV. OUR MOST EFFICIENT CONSTRUCTION

In this section, our main construction will be given where it will be provably secure under a concrete, non-interactive assumption. Moreover the scheme not only realizes expressive functionality but also gives an efficient decryption algorithm. The encryption algorithm in the proposed scheme will take a *LSSS* access matrix A as input and choose a set of random exponents from the distinct subgroups of $\mathbb{G}$. Therefore, private keys and ciphertexts in this scheme are randomized to protect the sensitive information of the access structure. Assume that $\mathscr{E} = (\mathscr{E}_{Enc}, \mathscr{E}_{Dec})$ is a symmetric encryption scheme with a message space $\mathscr{M}$.

Setup $(1^{\lambda}) \rightarrow PK, Msk$: The Setup algorithm takes a security parameter λ as input and outputs a tuple $(N = p_1p_2p_3p_4, \mathbb{G}, \mathbb{G}_T, e)$, where, (p_1, p_2, p_3, p_4) are distinct prime numbers. $\mathbb{G}_{p_1}, \mathbb{G}_{p_2}, \mathbb{G}_{p_3}, \mathbb{G}_{p_4}$ are distinct subgroups with order p_1, p_2, p_3, p_4, respectively. $\mathbb{G}$ and $\mathbb{G}_T$ are cyclic groups with order $N = p_1p_2p_3p_4$. The system uniformly chooses $a, \alpha, \alpha_1, \beta \in_R \mathbb{Z}_N$, and $g, g_1 \in \mathbb{G}_{p_1}$. Let H, H_1 be two public hash functions, where H maps an attribute value in VU_x to an element in $\mathbb{Z}_N$, and H_1 is a pseudo-random function which maps elements in $\mathbb{G}$ and $\mathscr{M}$ to elements in $\mathscr{M}$. e is a bilinear map: $\mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$. This algorithm computes $Y = e(g, g_1)^{\alpha\alpha_1}$, and sets the system public parameters and master key as $PK = \{N, g, g^a, g^{\alpha_1}, g^{\beta}, Y\}$, $Msk = \{a, \alpha, \alpha_1, \beta, g_1\}$, respectively.

KeyGen $(PK, MSK, S) \rightarrow SK$: The algorithm takes the public parameters PK, the master key MSK and the user's attributes set $S = (I_S, L_S)$ as input, where I_S denotes the user's attribute name index set and L_S is its attribute value set. It chooses $t \in_R \mathbb{Z}_N$ and $R, R_1, R_i \in_R \mathbb{G}_{P_3}$ for $i \in I_S$. The secret keys $SK = (K_1, K_2, \{K_i\}_{i \in I_S})$ associated with the attribute set $(S = I_S, L_S)$ are calculated as

$$
\begin{aligned}
K_1 &= g_1^{\alpha} {g_1}^{at} \cdot R \\
K_2 &= {g_1}^{t\alpha_1} \cdot R_1 \\
K_i &= ({g_1}^{H(L_i)} {g_1}^{\beta})^t \cdot R_i
\end{aligned}
$$

Encrypt $(PK, M, \mathbb{A}) \rightarrow CT$: It takes the message M, $\mathbb{A} = ((A, \rho), \mathcal{T})$ as input, where A is an $l \times n$ matrix and ρ is a map from each row A_x of access matrix A to an attribute name index. And $\mathcal{T} = (t_{\rho_{(1)}}, t_{\rho_{(2)}}, t_{\rho_{(3)}}, \ldots, t_{\rho_{(l)}}) \in \mathbb{Z}^l{}_N (t_{\rho_{(x)}} \in VU_{\rho_{(x)}})$ is the attribute value set associated with the access policy (A, ρ). This algorithm chooses a random vector $V = (s, y_2, y_3, \ldots, y_n)$, where $s, y_2, y_3, \ldots, y_n$ are chosen from $\mathbb{Z}_N$ at random and s is the shared secret value. For $x = 1$ to l, it calculates $\lambda_x = A_x \times V$, where A_x is the vector corresponding to the x^{th} row of A and computes $\mathcal{X} = \mathscr{E}_{Enc}(\mathcal{K}, M), \mathcal{F} = H_1(\mathcal{K}||M)$. In addition, it also chooses $Q_0, \{Q_x\}_{1 \le x \le l} \in_R \mathbb{G}_{P_4}$ uniformly at random. Finally, it calculates the rest of ciphertext components $(C_0, C_1, \{C_x\}_{1 \le x \le l})$ as follows:

$$
\begin{aligned}
C_0 &= \mathcal{K} e(g, g_1)^{\alpha \alpha_1 \, s}, \\
C_1 &= g^{s\alpha_1} \cdot Q_0, \\
C_x &= g^{a\lambda_x} (g^{H(t_{\rho(x)})} g^{\beta})^s \cdot Q_x.
\end{aligned}
$$

Decrypt $(PK, SK, CT, S) \rightarrow M$: After receiving $CT = ((A, \rho), C_0, C_1, \{C_x\}_{1 \le x \le l}, \mathcal{F})$, the algorithm first computes the set $\mathcal{I}$ by the access matrix (A, ρ) and the set I_S of decryption user, and then computes the constants $\{\omega_x\}_{x \in \mathcal{I}}$. If a decryption user is authorized, the values of $H(L_{\rho_{(x)}})$ and $H(t_{\rho_{(x)}})$ are equal, then the components about attributes from the ciphertexts and keys can be eliminated. So, it can reconstruct the secret value s hidden in the ciphertext by $\sum_{x \in I} \omega_x \lambda_x$. The decryption proceeds with $SK = (I_S, K_1, K_2, \{K_i\}_{i \in I_S})$ as follows:

$$
\begin{aligned}
E &= \frac{e(C_1, K_1 \prod_{x \in \mathcal{I}} K_{\rho(x)}{}^{\omega_x})}{e(\prod_{x \in \mathcal{I}} C_x{}^{\omega_x}, K_2)} = e(g, g_1)^{\alpha_1 \alpha s} \\
\mathcal{K} &= C_0 / E
\end{aligned}
$$

It is obvious that the plaintext M can only be recovered from equation $M' = \mathscr{E}_{Dec}(\mathcal{K}, \mathcal{X})$ if the key $\mathcal{K}$ of the symmetric encryption scheme is calculated. Then, the decryption algorithm calculates the value of $\mathcal{F}'$ by $H_1(\mathcal{K}'||M')$. If the equation $\mathcal{F}' = \mathcal{F}$ holds, it outputs M, otherwise, outputs $\perp$.

Correctness: The correctness of the decryption is given as follows:

$$
\begin{aligned}
\Delta_1 &= e(C_1, K_1 \prod_{x \in \mathcal{I}} K_{\rho(x)}{}^{\omega_x}) \\
&= e(g^{s\alpha_1} Q_0, g_1{}^{\alpha} g_1{}^{at} R \cdot \prod_{x \in \mathcal{I}} K_{\rho(x)}{}^{\omega_x}) \\
&= e(g^{s\alpha_1} Q_0, g_1{}^{\alpha} g_1{}^{at} R) \cdot e(g^{s\alpha_1} Q_1, g_1{}^{t \sum_{x \in \mathcal{I}} H(L_{\rho(x)}) \omega_x} \\
&\quad \times g_1{}^{t\beta \sum_{x \in \mathcal{I}} \omega_x} (\prod_{x \in \mathcal{I}} R_x)^{\omega_x})) \\
&= e(g^{s\alpha_1}, g_1{}^{\alpha}) \cdot e(g^{s\alpha_1}, g_1{}^{t \sum_{x \in \mathcal{I}} H(L_{\rho(x)}) \omega_x}) \\
&\quad \cdot e(g^{s\alpha_1}, g_1{}^{at}) \cdot e(g^{s\alpha_1}, g_1{}^{\beta t \sum_{x \in \mathcal{I}} \omega_x}) \\
\Delta_2 &= e(\prod_{x \in \mathcal{I}} C_x{}^{\omega_x}, K_2) \\
&= e(\prod_{x \in \mathcal{I}} (g^{a\lambda_x} (g^{H(t_{\rho(x)})} g^{\beta})^s \cdot Q_x)^{\omega_x}, g_1^{\alpha_1 t} \cdot R_1) \\
&= e(g^{as} \cdot g^{s \sum_{x \in \mathcal{I}} \omega_x H(t_{\rho(x)})} \cdot Q_x{}^{\sum_{x \in \mathcal{I}} \omega_x}, g_1^{\alpha_1 t} \cdot R_1) \\
&= e(g^{as}, g_1{}^{\alpha_1 t}) \cdot e(g^{s \sum_{x \in \mathcal{I}} \omega_x H(t_{\rho(x)})}, g_1^{\alpha_1 t}) \\
&\quad \cdot e(g^{s\beta \sum_{x \in \mathcal{I}} \omega_x}, g_1{}^{\alpha_1 t}),
\end{aligned}
$$

then

$$
\begin{aligned}
\frac{\Delta_1}{\Delta_2} &= \frac{e(g^{s\alpha_1}, g_1^{\alpha}) \cdot e(g^{s\alpha_1}, g_1^{t \sum_{x \in \mathcal{I}} H(L_{\rho(x)}) \omega_x})}{e(g^{s \sum_{x \in \mathcal{I}} \omega_x H(t_{\rho(x)})}, g_1^{\alpha_1 t})} \\
&= e(g, g_1)^{\alpha_1 \alpha s}
\end{aligned}
$$

From the above formulas, if and only if the decryption user satisfying the access structure (A, ρ), *i.e* $(H(L_{\rho(x)}) = H(t_{\rho(x)}))$, it can acquire the result $E = \Delta_1 / \Delta_2 = e(g, g_1)^{\alpha_1 \alpha s}$. Then, the value $\mathcal{K}$ can be correctly recovered by the equation $\mathcal{K} = C_0 / E$ and the message M can also be successfully reconstructed.

V. SECURITY PROOF

A. SEMI-FUNCTIONAL CIPHERTEXT AND SECRET KEY

Our security proof employs the approach as same as Lai's [5], which is called dual system encryption. At first, we define two semi-functional structures: *Semi-Functional Ciphertexts SFC* and *Semi-Functional Keys SFK*. Both normal ciphertexts and semi-functional ciphertexts can be decrypted by the normal private keys, but it is infeasible for a semi-functional private key to decrypt a semi-functional ciphertext. We made it clear in particular that *SFC* and *SFK* will not be used in the real system, and they only used in our proof.

1) SEMI-FUNCTIONAL CIPHERTEXT

We first set $CT' = (C_0', C_1', \{C_x'\}_{1 \le x \le l}, \mathcal{F}, \mathcal{X})$ as the normal ciphertext, which are encrypted by the normal encryption algorithm. Select a generator g_2 of group $\mathbb{G}_{P_2}$ and choose random exponents $f, u, \lambda_x, b, \mu_x \in \mathbb{Z}_N$, where, the value of μ_x corresponding to a certain attribute. The semi-functional ciphertext CT is set to be

$$
\begin{aligned}
C_0 &= C_0' = \mathcal{K} e(g_1, g)^{\alpha_1 \alpha s} \\
C_1 &= C_1' \cdot g_2^f \\
C_x &= C_x' \cdot g_2^{u\lambda_x + \eta \mu_x}
\end{aligned}
$$

2) SEMI-FUNCTIONAL KEY

In our scheme, a semi-functional key will employ one of three forms. We first set $SK' = (K_1', K_2', \{K_i'\}_{i \in I_S})$ as the normal secret key, and choose random exponents $d_1', d_2', \{\widehat{d_i}'\}_{i \in I_S} \in \mathbb{Z}_N$. Three types of semi-functional keys are set as follows:

Type 1:$(K_1 = K_1' \cdot g_2^{d_1'}, K_2 = K_2' \cdot g_2^{\alpha_1 d_2'}, \{K_i = K_i' \cdot g_2^{d_2' \pi_i}\}_{i \in I_S})$

Type 2:$(K_1 = K_1' \cdot g_2^{d_1'}, K_2 = K_2', \{K_i = K_i'\}_{i \in I_S})$

Type 3:$(K_1 = K_1' \cdot g_2^{d_1'}, K_2 = K_2' \cdot g_2^{\alpha_1 d_2'}, \{K_i = K_i' \cdot g_2^{\widehat{d_i}'}\}_{i \in I_S})$

The security proofs are based on *Assumptions* 1, 2, 3 and 4 by using a hybrid argument over a sequence of games. Now we define a series of games as follows:

Game$_0$: It is the first game, and its ciphertexts and all the keys are normal.

$Game_r$: As the second game, the challenge ciphertexts are set to semi-functional and all the keys are set to normal. In addition, $Game_r$ is also denoted as $Game_{0,3}$.

Let q denote the number of key queries between the challenger and adversary. And we define the following series of games, where $k \in [1, q]$.

$Game_{k,1}$: In this game, the challenge ciphertexts are generated by running the semi-functional encryption algorithm. The first $k-1$ keys are semi-functional of *Type 3*, and the k^{th} key is semi-functional of *Type 1* while the remaining keys are normal.

$Game_{k,2}$: In this game, the challenge ciphertexts are semi-functional, and the first $k-1$ keys are semi-functional of *Type 3*. The k^{th} key is semi-functional of *Type 2* while the remaining keys are normal.

$Game_{k,3}$: In this game, the challenge ciphertexts are set to semi-functional. The first k keys are set to semi-functional of *Type 3*, and the remaining keys are normal. But we need to emphasize that $Game_{q,3}(i.e, k = q)$, in which all the keys are semi-functional of *Type 3* and the ciphertexts are set to semi-functional.

$Game_{F_0}$: The challenge ciphertexts are the semi-functional encryption of a random message, in this game, independent of M_0 and M_1 chosen by the adversary $\mathscr{A}$, and all its keys are set to semi-functional of *Type 3*.

$Game_{F_1}$: This game is the same as $Game_{F_0}$, except that C_x in the challenge ciphertext is random elements in $\mathbb{G}_{P_1} \times \mathbb{G}_{P_2} \times \mathbb{G}_{P_4}$. And the challenge ciphertext is independent of $\mathcal{T}_0$ and $\mathcal{T}_1$ chosen by the adversary, therefore, the advantage of the adversary is 0.

B. PROOF PROCESS

Lemma 1: Suppose that the algorithm $\mathscr{G}$ satisfies *Assumption 1*. Then $Game_r$ and $Game_0$ are computationally indistinguishable.

Proof: If there exists an algorithm $\mathscr{A}$ such that $|Game_r Adv_{\mathscr{A}} - Game_0\ Adv_{\mathscr{A}}| = \epsilon$, where ϵ is non-negligible value. Then the algorithm $\mathscr{B}$ can be constructed with advantage ϵ in breaking *Assumption 1*. $\mathscr{B}$ is given a tuple (e, g, X_3, X_4, T) and will simulate $Game_r$ or $Game_0$ with $\mathscr{A}$. $\mathscr{B}$ uniformly chooses $a, \alpha, \alpha_1, \beta \in \mathbb{Z}_N$, $g, g_1 \in \mathbb{G}_{p_1}$ and sends $\mathscr{A}$ the system public key $PK = (e(g, g_1)^{\alpha}, g^{\beta}, g^{\alpha_1}, g^{a}, g)$. Meanwhile, $\mathscr{B}$ secretly keeps the master key $MSK = \{a, \alpha, \alpha_1, \beta, g_1\}$. When the adversary $\mathscr{A}$ requests the challenger about the private keys, and $\mathscr{B}$ can generate normal keys by using the key generation algorithm, since it knows the MSK. Then $\mathscr{A}$ sends $\mathscr{B}$ two challenge messages M_0 and M_1, ($|M_0| = |M_1|$), and two challenge access structures $(A, \rho, \mathcal{T}_0)$,$(A, \rho, \mathcal{T}_1)$ with restriction that none of them can be satisfying any of the queried attribute set in *phase 1*. In response, $\mathscr{B}$ flips a random coin $b \leftarrow \{0, 1\}$, chooses $Q_0, Q_x \in \mathbb{G}_{p_4}$ randomly. Furthermore, it also sets a column vector $\widetilde{V} = (1, \widetilde{y}_2, \widetilde{y}_3, \ldots, \widetilde{y}_n)$, the set of attribute value $\mathcal{T}_b = (t_{\rho_{(1)}}, t_{\rho_{(2)}}, \ldots, t_{\rho_{(l)}})$, and computes $\mathcal{X} = \mathscr{E}_{Enc}(\mathcal{K}, M_b)$, $\mathcal{F} = H_1(\mathcal{K}||M_b)$, $\widetilde{\lambda}_x = A_x \times \widetilde{V}$ where, $\{\widetilde{y}_2, \widetilde{y}_3, \ldots, \widetilde{y}_n\} \in \mathbb{Z}_N$. Then it does the following calculations.

$$
\begin{aligned}
C_0 &= \mathcal{K}e(g_1^{\alpha\alpha_1}, T),\\
C_1 &= T^{\alpha_1} Q_0,\\
C_x &= T^{a\widetilde{\lambda}_x} T^{(H(t_{\rho_{(x)}})+\beta)} Q_x.
\end{aligned}
$$

$\mathscr{B}$ sets the challenge ciphertext as $CT = \{C_0, C_1, C_x, \mathcal{F}, \mathcal{X}\}$, and sends CT to the adversary $\mathscr{A}$. If $T \overset{R}{\leftarrow} \mathbb{G}_{p_1} \times \mathbb{G}_{p_2}$, let $T = g^s g_2^c$, then

$$
\begin{aligned}
C_0 &= \mathcal{K}e(g_1^{\alpha\alpha_1}, g^s g_2^c) = \mathcal{K}e(g_1, g)^{s\alpha\alpha_1}\\
C_1 &= (g^s g_2^c)^{\alpha_1} Q_0 = g^{s\alpha_1} Q_0 \cdot {g_2}^{c\alpha_1}\\
C_x &= (g^s g_2^c)^{a\widetilde{\lambda}_x} (g^s g_2^c)^{(H(t_{\rho_{(x)}})+\beta)} Q_x\\
&= g^{as\widetilde{\lambda}_x} g^{s(H(t_{\rho_{(x)}})+\beta)} Q_x \cdot g_2^{ca\lambda_x} {g_2}^{c(H(t_{\rho_{(x)}})+\beta)}\\
&= g^{a\lambda_x} (g^{(H(t_{\rho_{(x)}})} g^{\beta})^s Q_x \cdot {g_2}^{m\lambda_x+\eta\mu_x}
\end{aligned}
$$

where $m = ca/s, \eta = c, f = c\alpha_1, \mu_x = H(t_{\rho_{(x)}}) + \beta, \lambda_x = s\widetilde{\lambda}_x$, and this is properly distributed since the values of $a, \alpha, \beta, \widetilde{y}_2, \widetilde{y}_2, \widetilde{y}_3, \ldots, \widetilde{y}_n$ modulo p_1 are uncorrelated from their values modulo p_2. Hence, this challenge ciphertext is *SFC* and $\mathscr{B}$ simulates the game $Game_0$. However, if $T \overset{R}{\leftarrow} \mathbb{G}_{p_1}$ then $\mathscr{B}$ simulates the game $Game_r$ and it is normal ciphertext. Hence, the challenger can distinguish between these possibilities for T by using the output of $\mathscr{A}$ and $Adv_1 G(\lambda) = \epsilon$.

Lemma 2: Suppose that the algorithm $\mathscr{G}$ satisfies *Assumption 2*. Then $Game_{k-1,3}$ and $Game_{k,1}$ are computationally indistinguishable.

Proof: If there exists an algorithm $\mathscr{A}$ such that $|Game_{k-1,3} Adv_{\mathscr{A}} - Game_{k,1} Adv_{\mathscr{A}}| = \epsilon$, then the algorithm $\mathscr{B}$ can be constructed with advantage ϵ in breaking *Assumption 2*. $\mathscr{B}$ is given a tuple $(X_1X_2, Y_2Y_3, X_3, X_4, T)$ and will simulate $Game_{k-1,3}$ or $Game_{k,1}$ with $\mathscr{A}$.

$\mathscr{B}$ uniformly chooses $a, \alpha, \alpha_1, \beta \in \mathbb{Z}_N$, $g, g_1 \in \mathbb{G}_{p_1}$ and sends $\mathscr{A}$ the system public key $PK = (g, g^a, g^{\alpha_1}, g^{\beta}, e(g, g_1)^{\alpha\alpha_1})$. Meanwhile, $\mathscr{B}$ secretly keeps the master key $MSK = \{a, \alpha, \alpha_1, \beta, g_1\}$. We now explain how $\mathscr{B}$ answers the j^{th} key query for attribute set $S = (I_S, L_S)$.

For $j > k$, it is normal key and $\mathscr{B}$ can create the key by running the normal key generation algorithm since it knows the master key.

For $j < k$, $\mathscr{B}$ uniformly chooses $t, \widetilde{d}_1, \widetilde{d}_2, \{\widehat{d}_i\}_{1 \le i \le |I_S|} \in \mathbb{Z}_N$, then, creates a semi-functional key of *Type 3* as follows:

$$
\begin{aligned}
K_1 &= {g_1}^{\alpha} {g_1}^{at} (Y_2Y_3)^{\widetilde{d}_1}\\
K_2 &= {g_1}^{\alpha_1 t} (Y_2Y_3)^{\widetilde{d}_2}\\
K_i &= ({g_1}^{H(L_i)} {g_1}^{\beta})^t (Y_2Y_3)^{\widehat{d}_i}
\end{aligned}
$$

Note that the values of $\widetilde{d}_1, \widetilde{d}_2, \{\widehat{d}_i\}_{1 \le i \le |I_S|}$ modulo p_2 are uncorrelated to their values modulo p_3, because this is a properly distributed semi-functional key of *Type 3*.

For $j = k$, $\mathscr{B}$ uniformly chooses $\{\overline{Z}, \widetilde{Z}_1, \widetilde{Z}_2, \{\widehat{Z}_i\}_{1 \le i \le |I_S|}$ $\in \mathbb{G}_{p_3}$ and sets

$$
K_1 = {g_1}^{\alpha} T^a \widetilde{Z}_1
$$

$$K_2 = T^{\alpha_1}\widetilde{Z}_2$$
$$K_i = T^{H(L_i)+\beta}\widehat{Z}_i$$

If $T \overset{R}{\leftarrow} \mathbb{G}_{P_1} \times \mathbb{G}_{P_2} \times \mathbb{G}_{P_3}$, without loss of generality, let $T = g_1^t g_2^{d_2'}\overline{Z}$, and computes

$$K_1 = {g_1}^{\alpha}(g_1^t g_2^{d_2'}\overline{Z})^a\widetilde{Z}_1 = g_1^{\alpha} g_1^{at}\overline{Z}^a\widetilde{Z}_1 \cdot g_2^{ad_2'}$$
$$K_2 = (g_1^t g_2^{d_2'}\overline{Z})^{\alpha_1}\widetilde{Z}_2 = g_1^{\alpha_1 t}\overline{Z}^{\alpha_1}\widetilde{Z}_2 \cdot g_2^{\alpha_1 d_2'}$$
$$K_i = (g_1^t g_2^{d_2'}\overline{Z})^{H(L_i)+\beta}\widehat{Z}_i$$
$$= g_1^{t(H(L_i)+\beta)}\overline{Z}^{H(L_i)+\beta}\widehat{Z}_i \cdot g_2^{d_2'(H(L_i)+\beta)}$$

We observe that $K_1 = g_1^{\alpha} g_1^{at} R \cdot {g_2}^{d_1'}$, $K_2 = g_1^t R_1 \cdot g_2^{d_2'}$, $\{K_i = (g_1^{H(L_i)} g_1^{\beta})^t R_i \cdot g_2^{d_2'\pi_i}\}_{1\le i\le |I_S|}$ where $R = \overline{Z}^a\widetilde{Z}_1$, $d_1' = ad_2'$, $R_1 = \overline{Z}^{\alpha_1}\widetilde{Z}_2$, $\{R_i = \overline{Z}^{H(L_i)+\beta}\widetilde{Z}_i\}_{1\le i\le |I_S|}$, $\pi_i = H(L_i) + \beta$. So this is a semi-functional key of *Type1*, since the values of $a, \alpha, \alpha_1, \beta$ modulo p_1 are uncorrelated to their values modulo p_2.

If $T \overset{R}{\leftarrow} \mathbb{G}_{P_1} \times \mathbb{G}_{P_3}$, it is a properly distributed normal key. When $\mathscr{A}$ submits to $\mathscr{B}$ two challenge messages M_0 and M_1, where M_0 and M_1 have the same length, and two challenge access structure $(A, \rho, \mathcal{T}_0)$,$(A, \rho, \mathcal{T}_1)$. $\mathscr{B}$ flips a random coin $b \leftarrow \{0, 1\}$, chooses $Q_1, \{Q_x\}_{1\le x\le l} \in \mathbb{G}_{p_4}$ and $\{\widetilde{y}_2, \widetilde{y}_3, \ldots, \widetilde{y}_n\} \in \mathbb{Z}_N$ randomly. Moreover, it also sets a column vector $\widetilde{V} = (1, \widetilde{y}_2, \widetilde{y}_3, \ldots, \widetilde{y}_n)$, calculates $\mathcal{X} = \mathscr{E}_{Enc}(\mathcal{K}, M_b)$, $\mathcal{F} = H_1(\mathcal{K}||M_b)$ and does the remaining calculation.

$$C_0 = \mathcal{K}e({g_1}^{\alpha\alpha_1}, X_1X_2)$$
$$C_1 = (X_1X_2)^{\alpha_1}Q_1$$
$$C_x = (X_1X_2)^{a\widetilde{\lambda}_x}(X_1X_2)^{(H(t_{\rho(x)})+\beta)}Q_x$$

$\mathscr{B}$ sets the challenge ciphertext as $CT = \{C_0, C_1, C_x, \mathcal{F}, \mathcal{X}\}$, and sends CT to the adversary $\mathscr{A}$. If $X_1X_2 = g^s g_2^c$, then

$$C_0 = \mathcal{X}e(g_1^{\alpha\alpha_1}, g^s g_2^c) = \mathcal{X}e(g_1, g)^{s\alpha\alpha_1}$$
$$C_1 = (g^s g_2^c)^{\alpha_1}Q_1 = g^{s\alpha_1}Q_1 \cdot g_2^{c\alpha_1}$$
$$C_x = (g^s g_2^c)^{a\widetilde{\lambda}_x}(g^s g_2^c)^{(H(t_{\rho(x)})+\beta)}Q_x$$
$$= g^{as\widetilde{\lambda}_x} g^{s(H(t_{\rho(x)})+\beta)}Q_x \cdot g_2^{ca\lambda_x} g_2^{c(H(t_{\rho(x)})+\beta)}$$
$$= g^{a\lambda_x}(g^{(H(t_{\rho(x)})}g^{\beta})^s Q_x \cdot g_2^{u\lambda_x+\eta\mu_x}$$

where $f = c\alpha_1$, $\mu_x = H(t_{\rho(x)}) + \beta$, $\lambda_x = s\widetilde{\lambda}_x$, $\eta = c$, $u = ca/s$. This is a semi-functional ciphertext, since the value of $a, \widetilde{y}_2, \widetilde{y}_3, \ldots, \widetilde{y}_n$ modulo p_1 are uncorrelated to their values modulo p_2. Therefore, if $T \overset{R}{\leftarrow} \mathbb{G}_{p_1} \times \mathbb{G}_{p_2} \times \mathbb{G}_{p_3}$, then $\mathscr{B}$ has properly simulated $Game_{k,1}$. If $T \overset{R}{\leftarrow} \mathbb{G}_{p_1} \times \mathbb{G}_{p_3}$, then $\mathscr{B}$ has properly simulated $Game_{k-1,3}$. Hence, the challenger can distinguish between these possibilities for T by using the output of $\mathscr{A}$ and $Adv_2G(\lambda) = \epsilon$.

Lemma 3: Suppose that the algorithm $\mathscr{G}$ satisfies *Assumption 2*. Then $Game_{k,1}$ and $Game_{k,2}$ are computationally indistinguishable.

Proof: If there exists an algorithm $\mathscr{A}$ such that $|Game_{k,1}Adv_{\mathscr{A}} - Game_{k,2}Adv_{\mathscr{A}}| = \epsilon$, then the algorithm $\mathscr{B}$ can be constructed with advantage ϵ in breaking *Assumption 2*. $\mathscr{B}$ is given a tuple $(g, X_1X_2, Y_2Y_3, X_3, X_4, T)$ and it will simulate $Game_{k,1}$ or $Game_{k,2}$ with $\mathscr{A}$.

$\mathscr{B}$ uniformly chooses $\alpha, a, \alpha_1, \beta \in \mathbb{Z}_N$, $g, g_1 \in \mathbb{G}_{p_1}$ and sends $\mathscr{A}$ the public parameters $PK = (g, g^{\beta}, g^a, g^{\alpha_1}, e(g, g_1)^{\alpha\alpha_1})$. Moreover, $\mathscr{B}$ secretly keeps the master key $MSK = \{a, \alpha, \beta, \alpha_1, g_1\}$. We now explain how $\mathscr{B}$ answers the j^{th} key query for attribute set $S = (I_S, L_S)$. Except for the k^{th} semi-functional key, the first $k-1$ semi-functional keys of *Type 3* and the process of generating are as same as *Lemma 2*. So the remaining keys can be generated by running the normal algorithm and we do not repeat them here.

To answer the k^{th} key request for $S = (I_S, L_S)$, $\mathscr{B}$ uniformly chooses $\widehat{r} \in \mathbb{Z}_N$ and sets

$$K_1 = g^{\alpha}T^a\widetilde{Z}_1 \cdot (Y_2Y_3)^{\widehat{r}}$$
$$K_2 = T^{\alpha_1}\widetilde{Z}_2$$
$$K_i = T^{H(L_i)+\beta}\widetilde{Z}_i$$

From the above formula, it is obvious that the only change in the key is the adding the $(Y_2Y_3)^{r'}$ term, which randomizes the $\mathbb{G}_{P_2}$ part of K_1. If $T \overset{R}{\leftarrow} \mathbb{G}_{p_1} \times \mathbb{G}_{p_2} \times \mathbb{G}_{p_3}$, where T is made up of three subgroups' elements of $\mathbb{G}_{p_1}, \mathbb{G}_{p_2}$ and $\mathbb{G}_{p_3}$, respectively. So the *SFK* can be guaranteed to be consistent with key of *Type 1*, and then, $\mathscr{B}$ has simulated the game $Game_{k,1}$. If $T \overset{R}{\leftarrow} \mathbb{G}_{p_1} \times \mathbb{G}_{p_3}$, it is a correctly simulated semi-functional key of *Type 2*, and $\mathscr{B}$ has simulated the game $Game_{k,2}$. Hence, the challenger can distinguish between these possibilities for T by using the output of $\mathscr{A}$ and $Adv_2G(\lambda) = \epsilon$.

Lemma 4: Suppose that the algorithm $\mathscr{G}$ satisfies *Assumption 2*. Then $Game_{k,2}$ and $Game_{k,3}$ are computationally indistinguishable.

Proof: If there exists an algorithm $\mathscr{A}$ such that $|Game_{k,2}Adv_{\mathscr{A}} - Game_{k,3}Adv_{\mathscr{A}}| = \epsilon$, then the algorithm $\mathscr{B}$ can be constructed with advantage ϵ in breaking *Assumption 2*. $\mathscr{B}$ is given a tuple $(g, X_1X_2, Y_2Y_3, X_3, X_4, T)$ and it will simulate $Game_{k,2}$ or $Game_{k,3}$ with $\mathscr{A}$.

$\mathscr{B}$ randomly chooses $\alpha, \alpha_1, a, \beta \in \mathbb{Z}_N$, $g, g_1 \in \mathbb{G}_{p_1}$ and sends $\mathscr{A}$ the system public key $PK = (g, g^{\beta}, g^a, g^{\alpha_1}, e(g, g_1)^{\alpha\alpha_1})$. $\mathscr{B}$ secretly keeps the master key $MSK = \{a, \alpha, \alpha_1, \beta, g_1\}$. When the adversary requests the challenger about the key. Except for the k^{th} semi-functional key, the first $k-1$ semi-functional keys of *Type 3* and the process of generating are as same as *Lemma 2*. And the remaining keys are normal, which can be generated by running the normal algorithm and we are not repeated here. All the challenge ciphertext can also be constructed in the same way as in *Lemma 2*. So we now explain how $\mathscr{B}$ answers the k^{th} key query for attribute set $S = (I_S, L_S)$.

In order to answer the k^{th} key quest for $S = (I_S, L_S)$, $\mathscr{B}$ randomly chooses $r, r_1, \widehat{t}, \widehat{d} \in \mathbb{Z}_N$ and $\{\overline{Z}, \widetilde{Z}_1, \widetilde{Z}_2, \{\widehat{Z}_i\}_{1\le i\le |I_S|}\} \in \mathbb{G}_{p_3}$, sets

$$K_1 = {g_1}^{\alpha}T^{ra}\widetilde{Z}_1 \cdot (Y_2Y_3)^{r_1}$$
$$K_2 = T^{r\alpha_1}\widetilde{Z}_2$$
$$K_i = T^{r(H(L_x)+\beta)}\widetilde{Z}_i.$$

If $T \overset{R}{\leftarrow} \mathbb{G}_{p_1} \times \mathbb{G}_{p_2} \times \mathbb{G}_{p_3}$, set $T = g_1^{\widehat{t}} g_2^{\widehat{d}} \overline{Z}$ and do the following calculations.

$$\begin{aligned} K_1 &= g_1{}^{\alpha}(g_1^{\widehat{t}} g_2^{\widehat{d}} \overline{Z})^{a} \widetilde{Z}_1 \cdot (Y_2 Y_3)^{r_1} \\ &= g_1^{\alpha} g_1^{ar\widehat{t}} (\widetilde{Z}_1 \overline{Z}^{ar} Y_3^{r_1}) \cdot (g_2^{ar\widehat{d}} Y_2^{r_1}) \\ K_2 &= (g_1^{\widehat{t}} g_2^{\widehat{d}} \overline{Z})^{\alpha_1 r} \widetilde{Z}_2 \\ &= g_1^{\alpha_1 r\widehat{t}} \overline{Z}^{\alpha_1 \widehat{r}} \widetilde{Z}_2 \cdot g_2^{\widehat{r}\alpha_1 \widehat{d}} \\ K_i &= (g_1^{\widehat{t}} g_2^{\widehat{d}} \overline{Z})^{r(H(L_i)+\beta)} \widetilde{Z}_i \\ &= (g_1^{H(L_i)+\beta})^{\widehat{t}r} \overline{Z}^{r(H(L_i)+\beta)} \widetilde{Z}_i \cdot g_2^{r(H(L_i)+\beta)\widehat{d}} \end{aligned}$$

We observe that $K_1 = K_1' \cdot g_2^{d_1'}$, $K_2 = K_2' \cdot g_2^{d_2'}$, $\{K_i = K_i' \cdot g_2^{\widehat{d}_i'}\}_{1 \le i \le |I_S|}$, where $R = \overline{Z}^{ar} \widetilde{Z}_1 Y_3{}^{r_1}$, $g_2{}^{d_1'} = g_2{}^{ar\widehat{d}} Y_2^{r_1}$, $R_1 = \overline{Z}^{\alpha_1 r} \widetilde{Z}_2$, $d_2' = \alpha_1 r\widehat{d}$, $\{R_i = \overline{Z}^{r(H(L_i)+\beta)} \widetilde{Z}_i\}_{1 \le i \le |I_S|}$, $\{\widehat{d}_i' = r(H(L_i)+\beta)\widehat{d}\}_{1 \le i \le |I_S|}$, so it is a semi-functional key of *Type 3*, since the values of r, r_1 is chosen from $\mathbb{Z}_N$ randomly, so r and r_1 modulo p_2 are uncorrelated to their values modulo p_3.

If $T \overset{R}{\leftarrow} \mathbb{G}_{p_1} \times \mathbb{G}_{p_3}$, this is a correctly simulated semi-functional key of *Type 2*, and then, $\mathscr{B}$ has simulated the game $Game_{k,2}$. If $T \overset{R}{\leftarrow} \mathbb{G}_{p_1} \times \mathbb{G}_{p_2} \times \mathbb{G}_{p_3}$, $\mathscr{B}$ has simulated the game $Game_{k,3}$. Hence, the challenger can distinguish between these possibilities for T by using the output of $\mathscr{A}$ and $Adv_2 G(\lambda) = \epsilon$.

Lemma 5: Suppose that the algorithm $\mathscr{G}$ satisfies *Assumption 3*. Then $Game_{q,3}$ and $Game_{F_0}$ are computationally indistinguishable.

Proof: If there exists an algorithm $\mathscr{A}$ such that $|Game_{q,3}Adv_{\mathscr{A}} - Game_{F_0}Adv_{\mathscr{A}}| = \epsilon$, then the algorithm $\mathscr{B}$ can be constructed with advantage ϵ in breaking *Assumption 3*. $\mathscr{B}$ is given a tuple $(g, g_2, g_\alpha X_2, g^s Y_2, X_3, X_4, T)$ and it will simulate $Game_{q,3}$ or $Game_{F_0}$ with $\mathscr{A}$.

$\mathscr{B}$ uniformly chooses $\alpha, \alpha_1, a, \beta \in \mathbb{Z}_N$, $g, g_1 \in \mathbb{G}_{p_1}$ and sends $\mathscr{A}$ the system public key $PK = (g, g^{\beta}, g^{\alpha_1}, g^{a}, e(g, g_1)^{\alpha\alpha_1})$. Meanwhile $\mathscr{B}$ secretly keeps the master key $MSK = \{a, \alpha, \alpha_1, \beta, g_1\}$. When the adversary $\mathscr{A}$ requests the challenger about the key, $\mathscr{B}$ randomly chooses $t \in \mathbb{Z}_N$, $\widehat{d}_1 \in \mathbb{Z}_N$ to create a semi-functional key of *Type 3* and set

$$\begin{aligned} K_1 &= (g_1{}^{\alpha} X_2) g_1{}^{at} R \cdot g_2{}^{\widehat{d}_1} \\ K_2 &= g_1{}^{\alpha_1 t} R_1 \cdot g_2{}^{d_2'} \\ K_i &= (g_1{}^{H(L_i)} g_1{}^{\beta})^{t} R_i \cdot g_2{}^{\widehat{d}_i'} \end{aligned}$$

where $g_2{}^{d_1'} = g_2{}^{\widehat{d}_1} X_2$. Note that $K_1 = g_1^{\alpha} g_1^{at} R \cdot g_2{}^{\widehat{d}_1}$, so this is a properly distributed semi-functional key of *Type 3*. When $\mathscr{A}$ submits to $\mathscr{B}$ two challenge messages M_0 and M_1, where ($|M_0| = |M_1|$), and two challenge access structures $(A, \rho, \mathcal{T}_0)$, $(A, \rho, \mathcal{T}_1)$. The restriction that none of these two access structures submitted by $\mathscr{A}$ satisfying any of the queried attribute sets. $\mathscr{B}$ chooses $b \leftarrow \{0, 1\}$, $Q_0, Q_x \in \mathbb{G}_{p_4}$ and $\{\widetilde{y}_2, \widetilde{y}_3, \ldots, \widetilde{y}_n\} \in \mathbb{Z}_N$ randomly. $\mathscr{B}$ sets a column vector $\widetilde{V} = (1, \widetilde{y}_2, \widetilde{y}_3, \ldots, \widetilde{y}_n)$, and computes $\mathcal{X} = \mathscr{E}_{Enc}(\mathcal{K}, M_b)$, $\mathcal{F} = H_1(\mathcal{K} || M_b)$, $\widetilde{\lambda}_x = A_x \widetilde{V}$. The challenge ciphertexts are set as follows:

$$\begin{aligned} C_0 &= \mathcal{K}T \\ C_1 &= (g^s Y_2)^{\alpha_1} Q_0 \\ C_x &= (g^s Y_2)^{a\widetilde{\lambda}_x} (g^s Y_2)^{H(t_{\rho(x)})+\beta} Q_x \end{aligned}$$

$\mathscr{B}$ sets the challenge ciphertext as $CT = \{C_0, C_1, C_x, \mathcal{F}, \mathcal{X}\}$, and sends CT to $\mathscr{A}$. Let $g^s Y_2 = g^s g_2^c$ then

$$\begin{aligned} C_0 &= \mathcal{K}T \\ C_1 &= (g^s g_2^c)^{\alpha_1} Q_0 = g^{\alpha_1 s} Q_0 \cdot g_2^{\alpha_1 c} \\ C_x &= (g^s g_2^c)^{a\widetilde{\lambda}_x} (g^s g_2^c)^{H(t_{\rho(x)})+\beta)} Q_x \\ &= g^{sa\widetilde{\lambda}_x} g^{s(H(t_{\rho(x)})+\beta)} \cdot g_2^{c(a\widetilde{\lambda}_x + (H(t_{\rho(x)})+\beta))} \\ &= g^{a\lambda_x} (g^{H(L_x)} g^{\beta})^{s} Q_x \cdot g_2^{m\lambda_x + \eta\mu_x} \end{aligned}$$

where $f = c\alpha_1$, $\lambda_x = s\widetilde{\lambda}_x$, $m = ca/s$, $\eta = c$, $\mu_x = H(t_{\rho(x)})+\beta$. If $T = e(g, g_1)^{\alpha\alpha_1 s}$, this is a correctly distributed semi-functional encryption of M_b, and then, $\mathscr{B}$ simulates the game $Game_{q,3}$. Otherwise, it is a properly distributed semi-functional encryption of a random message in $\mathbb{G}_T$, and $\mathscr{B}$ simulates the game $Game_{F_0}$. Hence, the challenger $\mathscr{B}$ can distinguish between these possibilities for T by using the output of $\mathscr{A}$ and $Adv_3 G(\lambda) = \epsilon$.

Lemma 6: Suppose that the algorithm $\mathscr{G}$ satisfies *Assumption 4*. Then $Game_{F_0}$ and $Game_{F_1}$ are computationally indistinguishable.

Proof: If there exists an algorithm $\mathscr{A}$ such that $|Game_{F_0}Adv_{\mathscr{A}} - Game_{F_1}Adv_{\mathscr{A}}| = \epsilon$, then the algorithm $\mathscr{B}$ can be constructed with advantage ϵ in breaking *Assumption 4*. $\mathscr{B}$ is given a tuple $(g, g_2, g_1^{t'} B_2, X_3, X_4, g^{r'} D_2 D_4, T)$ and it will simulate $Game_{F_0}$ or $Game_{F_1}$ with $\mathscr{A}$.

$\mathscr{B}$ uniformly chooses $\alpha, \alpha_1, a, \beta \in \mathbb{Z}_N$,$g, g_1 \in \mathbb{G}_{p_1}$ and acquires the public parameters and master key by running the algorithm ***Setup***. Then, $\mathscr{B}$ sends $\mathscr{A}$ the system public key $PK = (N, H_1, H_2, g, g^{\beta}, g^{\alpha_1}, g^{a}, e(g, g_1)^{\alpha\alpha_1})$ and secretly keeps the master key $MSK = \{a, \alpha, \alpha_1, \beta, g_1\}$. When the adversary $\mathscr{A}$ requests the challenger about the key, $\mathscr{B}$ uniformly chooses $\bar{t} \in \mathbb{Z}_N$, $R, R_1, \{R_i\} \in \mathbb{G}_{p_3}$ for $1 \le i \le |I_S|$, and creates a semi-functional key of *Type 3* as follows:

$$\begin{aligned} K_1 &= g_1^{\alpha} (g_1^{t'} B_2)^{a\bar{t}} = g_1^{\alpha} g_1^{a\bar{t}t'} R \cdot B_2^{a\bar{t}} \\ K_2 &= (g_1^{t'} B_2)^{\alpha_1 \bar{t}} \cdot R_1 = g_1^{\alpha_1 \bar{t}t'} R_1 \cdot B_2^{\alpha_1 \bar{t}} \\ K_i &= (g_1^{t'} B_2)^{\bar{t}H(L_i)} (g_1^{t'} Y_2)^{\beta\bar{t}} \cdot R_i \\ &= g_1^{\bar{t}t'H(L_i)} g_1^{\beta\bar{t}t'} R_i \cdot B_2^{\bar{t}H(L_i)} Y_2^{\bar{t}\beta}. \end{aligned}$$

We observe that $K_1 = K_1' g_2^{d_1'}$, $K_2 = K_2' g_2^{d_2'}$, $\{K_i = K_i' g_2^{\widehat{d}_i'}\}_{1 \le i \le |I_S|}$ where $t = \bar{t}t'$, $g_2^{d_1'} = B_2^{a\bar{t}}$, $g_2^{d_2'} = B_2^{\alpha_1 \bar{t}}$, $g_2^{\widehat{d}_i'} = B_2^{\bar{t}H(L_i)} Y_2^{\bar{t}\beta}$. So it's a correctly distributed semi-functional key of *Type 3*. Then $\mathscr{A}$ submits to $\mathscr{B}$ two challenge messages M_0 and M_1 of the same length and two challenge access structures $(A, \rho, \mathcal{T}_0)$, $(A, \rho, \mathcal{T}_1)$. However, the restriction is that none of these two access structures submitted by $\mathscr{A}$ satisfying any of the queried attribute sets. $\mathscr{B}$ flips a random coin $b \leftarrow \{0, 1\}$, and also chooses $\widetilde{r}, r \in_R \mathbb{Z}_N$, $Q_0, \{\widetilde{Q}_x\}_{1 \le x \le l} \in \mathbb{G}_{p_4}$ and $y_2, y_3, \ldots, y_n \in \mathbb{Z}_N$ randomly. In addition, it also sets a column vector $V = (s, y_2, y_3, \ldots, y_n)$, the set of

TABLE 1. Comparisons with other schemes.

scheme	Size of PK	Size of SK	Size of CT	Dec. Cost	Group order	Access structure
Lai [5]	$(n+3)\|\mathbb{G}\|+2\|\mathbb{G}_T\|$	$(\|I\|+2)\|\mathbb{G}\|$	$(4l+2)\|\mathbb{G}\|+2\|\mathbb{G}_T\|$	$(2\|I\|+1)\mathbf{p}$	$p_1 \ldots p_4$	LSSS
Zhang [8]	$3\|\mathbb{G}\|+2\|\mathbb{G}_T\|$	$(\|I\|+2)\|\mathbb{G}\|$	$(3l+2)\|\mathbb{G}\|+2\|\mathbb{G}_T\|$	$(2\|I\|+1)\mathbf{p}$	$p_1 \ldots p_4$	LSSS
Cui [9]	$9\|\mathbb{G}\|+\|\mathbb{G}_T\|$	$(5\|I\|+2)\|\mathbb{G}\|$	$(6l+2)\|\mathbb{G}\|+\|\mathbb{G}_T\|$	$(6\|I\|+1)\mathbf{p}$	p	LSSS
Jin [18]	$(8n+3)\|\mathbb{G}\|+2\|\mathbb{G}_T\|$	$(4n+2)\|\mathbb{G}\|$	$(4l+2)\|\mathbb{G}\|+\|\mathbb{G}_T\|$	$(4\|I\|+2)\mathbf{p}$	$p_1 p_2 p_3$	AND on +/-
Wang [24]	$(n+3)\|\mathbb{G}\|+\|\mathbb{G}_T\|$	$(\|I\|+2)\|\mathbb{G}\|$	$(2l+1)\|\mathbb{G}\|+\|\mathbb{G}_T\|$	$(2\|I\|+1)\mathbf{p}$	$p_1 p_2 p_3$	LSSS
Ours	$4\|\mathbb{G}\|+\|\mathbb{G}_T\|$	$(\|I\|+2)\|\mathbb{G}\|$	$(l+1)\|\mathbb{G}\|+\|\mathbb{G}_T\|$	$2\mathbf{p}$	$p_1 \ldots p_4$	LSSS

attribute value $\mathcal{T}_b = (t_{\rho_{(1)}}, t_{\rho_{(2)}}, \ldots, t_{\rho_{(l)}})$, and calculates $\mathcal{X} = \mathscr{E}_{Enc}(\mathcal{K}, M_b)$, $\mathcal{F} = H_1(\mathcal{K}||M_b)$, $\lambda_x = A_x V$. Then, does the following calculation.

$$
\begin{aligned}
C_0 &\overset{R}{\leftarrow} \mathbb{G}_T \\
C_1 &= (g^s g_2^c)^{\alpha_1} \cdot Q_0 \\
C_x &= (g^{r'} D_2 D_4)^{a\lambda_x} T^{s(H(t_{\rho_{(x)}})+\beta)} \cdot \widetilde{Q}_x
\end{aligned}
$$

$\mathscr{B}$ sets the challenge ciphertext as $CT = \{C_0, C_1, C_x, \mathcal{X}, \mathcal{F}\}$, and sends CT to $\mathscr{A}$. If $T = g^{r'} A_2 A_4$,let $D_2 = g_2^{\widetilde{r}}$, $A_2 = g_2^{r\widetilde{r}}$, then

$$
\begin{aligned}
C_0 &\overset{R}{\leftarrow} \mathbb{G}_T \\
C_1 &= (g^s g_2^c)^{\alpha_1} \cdot Q_0 = g^{\alpha_1 s} Q_0 \cdot g_2^{c\alpha_1} \\
C_x &= (g^{r'} D_2 D_4)^{a\lambda_x} (g^{r'} A_2 A_4)^{s(H(t_{\rho_{(x)}})+\beta)} \cdot \widetilde{Q}_x \\
&= g^{r'\alpha\lambda_x} (g^{H(t_{(\rho_x)})} g^{\beta})^{(r's)} g_2^{\alpha\widetilde{r}\lambda_x + r\widetilde{r}s(H(t_{\rho(x)})+\beta)} \\
&\quad \cdot D_4^{a\lambda_x} A_2^{s\mu_x} \widetilde{Q}_x
\end{aligned}
$$

where $m = \alpha\widetilde{r}, \eta = \widetilde{r}rs, \mu_x = H(t_{\rho_{(x)}} + \beta), Q_x = D_4^{a\lambda_x} A_2^{s\mu_x} \widetilde{Q}_x$, it is a correctly distributed semi-functional encryption of a random message in $\mathbb{G}_T$, because the values of $r, \widetilde{r}, t_{\rho_{(x)}}$ modulo p_2 are uncorrelated to their values modulo p_4.

If $T \overset{R}{\leftarrow} \mathbb{G}_{p_1} \times \mathbb{G}_{p_2} \times \mathbb{G}_{p_4}$, this is a correctly distributed semi-functional ciphertext with C_0 random in $\mathbb{G}_T$ and C_x random in $\mathbb{G}_{p_1} \times \mathbb{G}_{p_2} \times \mathbb{G}_{p_4}$, so $\mathscr{B}$ has properly simulated the game $Game_{F_1}$. If $T = g^{r'} A_2 A_4$, then $\mathscr{B}$ has correctly simulated the game $Game_{F_0}$. Hence, the challenger $\mathscr{B}$ can distinguish between these possibilities for T by using the output of $\mathscr{A}$ and $Adv_4 G(\lambda) = \epsilon$.

VI. PERFORMANCE ANALYSIS

In this section, we will give some comparisons of our scheme with previous related works ([5], [8], [9], [18], [24]) in terms of security and performance. In Table 1, we provide comprehensive comparisons for some important features, including the size of public keys, private keys and ciphertexts, decryption overhead, group order, and the expression and status of access policy. From Table 1, we can see that the size of keys in our scheme is the same with other works, but the ciphertext size of proposed scheme is smaller than them. In addition, only the proposed scheme and the work in [8] support large universe constructions. What's more, compared with the above work, only our scheme can realize constant pairing operation in the decryption phase, which can greatly improve the decryption efficiency. In Table 1, $\mathbf{p}$ denotes the pairing operation. l and n are the size of access matrix and the category of attributes in the universes respectively. I is a set satisfying the access structure defined by an encryptor. $|\mathbb{G}|$ and $|\mathbb{G}_T|$ denote the number of bits for the representation of elements of $\mathbb{G}$ and $\mathbb{G}_T$. Note that the size of an element in group $\mathbb{G}_{p_i}$, $\mathbb{G}$ and $\mathbb{G}_T$ is set to 512 bits.

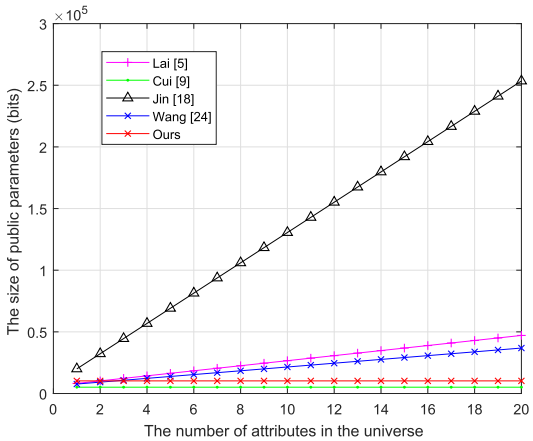

FIGURE 3. The storage cost of the public parameters.

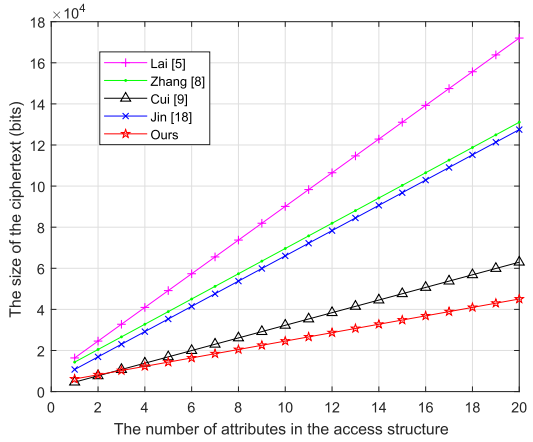

FIGURE 4. The storage cost of the ciphertext.

Fig.3, 4 and Fig.5 give comparisons of the performance advantage of the proposed scheme with the above. The simulation is accomplished on a Windows machine with 3.40 GHz Intel(R) Core(TM) i3-3240 CPU and 4 GB ROM. From the Fig.3, 4 and Fig.5, it is obvious that the proposed scheme is better than others in Table 1 in terms of the size of public parameters, ciphertexts and decryption overhead.

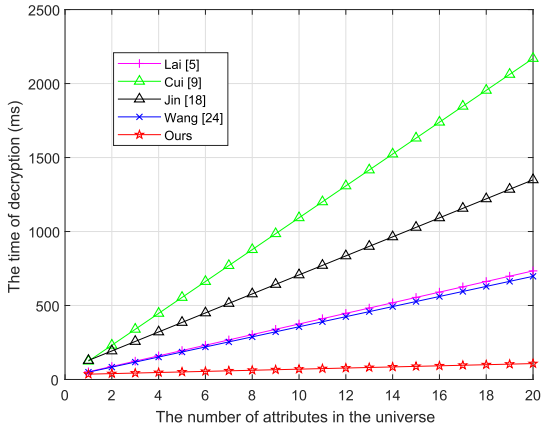

FIGURE 5. The decryption overhead for data users.

VII. CONCLUSION

In this paper, we introduce a new method called linear secret sharing with multiple values, which can greatly improve the expression of access policy. Moreover, each attribute is divided into two parts, namely the attribute name and its value. Therefore, the most obvious advantage of the proposed scheme is that sensitive attribute values can be hidden. And it can protect users' privacy well in PHR. In the proposed scheme, the size of public parameters is constant and the cost of the decryption is only two pairing operations, which also make it more practical. Eventually, we prove the full security of the proposed scheme in the standard model under static assumptions by using the dual system encryption method. The proposed scheme only achieves partly hiding policy. It is an interesting problem that achieves fully hiding policy with fast encryption, which is left as a future work.

REFERENCES

[1] B. Waters, "Dual system encryption: Realizing fully secure IBE and HIBE under simple assumptions," in *Proc. Annu. Int. Cryptol. Conf.* Berlin, Germany: Springer, Aug. 2009, pp. 619–636.
[2] M. Qutaibah, S. Abdullatif, and C. T. Viet, "A ciphertext-policy attribute-based encryption scheme with optimized ciphertext size and fast decryption," in *Proc. ACM Asia Conf. Comput. Commun. Secur.*, Apr. 2017, pp. 230–240.
[3] B. Waters, "Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization," in *Proc. Int. Workshop Public Key Cryptogr.* Berlin, Germany: Springer, Mar. 2011, pp. 53–70.
[4] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in *Proc. 13th ACM Conf. Comput, Commun. Secur.*, Nov. 2006, pp. 89–98.
[5] J. Lai, R. H. Deng, and Y. Li, "Expressive CP-ABE with partially hidden access structures," in *Proc. 7th ACM Symp. Inf., Comput. Commun. Secur.*, May. 2012, pp. 18–19.
[6] A. Sahai and B. Waters, *Fuzzy Identity-Based Encryption*, R. Cramer, Eds. Berlin, Germany: Springer, 2005, pp. 457–473.
[7] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attributebased encryption," in *Proc. IEEE Symp. Secur. Privacy*, May. 2007, pp. 321–334.
[8] Y. Zhang, D. Zheng, and R. H. Deng, "Security and privacy in smart health: Efficient policy-hiding attribute-based access control," *IEEE Internet Things J.*, vol. 5, no. 3, pp. 2130–2145, Jun. 2018.
[9] H. Cui, R. H. Deng, G. Wu, and J. Lai, "An efficient and expressive ciphertext-policy attribute-based encryption scheme with partially hidden access structures," in *Provable Security—PROVSEC* (Lecture Notes in Computer Science), vol. 10005, L. Chen, Eds. Berlin, Germany: Springer, 2016, pp. 19–38.
[10] C. Y. Umesh, "Ciphertext-policy attribute-based encryption with hiding access structure," in *Proc. IEEE Inter. Adv. Comput. Conf.*, Jul. 2015, pp. 6–10.
[11] L. Zhang and Y. Hu, "New constructions of hierarchical attribute-based encryption for fine-grained access control in cloud computing," *KSII Trans. Int. Information Syst.*, vol. 7, no. 5, pp. 1343–1356, May 2013.
[12] J. Li, K. Ren, B. Zhu, and Z. Wan, "Privacy-aware attribute-based encryption with user accountability," in *Information Security—PROCEEDINGS* (Lecture Notes in Computer Science) vol. 5735, P. Samarati, Eds. Berlin, Germany: Springer, 2009, pp. 347–362.
[13] J. Li, H. Wang, Y. Zhang, and J. Shen, "Ciphertext-policy attribute-based encryption with hidden access policy and testing," *KSII Trans. Int. Information Syst.*, vol. 10, no. 7, pp. 3339–3352, Jul. 2016.
[14] Y. Zhang, X. Chen, J. Li, and D. Wong, "Anonymous attribute-based encryption supporting efficient decryption test," in *Proc. 8th ACM SIGSAC Symp. Inf., Comput. Commun. Secur.*, May 2013, pp. 511–516.
[15] K. Emura, A. Miyaji, A. Nomura, K. Omote, and M. Soshi, "A ciphertext-policy attribute-based encryption scheme with constant ciphertext length," in *Information Security Practice and Experience—ISPEC* (Lecture Notes in Computer Science) vol. 5451, F. Bao, H. Li, Eds. Berlin, Germany: Springer, 2009, pp. 13–23.
[16] T. Nishide, K. Yoneyama, and K. Ohta, "Attribute-Based Encryption with Partially Hidden Encryptor-Specified Access Structures," in *Proc. Appl. Cryptography Netw. Security—ACNS* (Lecture Notes in Computer Science) vol. 5037, S. M. Bellovin, R. Gennaro, Eds. Berlin, Germany: Springer, 2009, pp. 13–23.
[17] T. V. X. Phuong, G. Yang, and W. Susilo, "Hidden ciphertext policy attribute-based encryption under standard assumptions," *IEEE Trans. Inf. Forensics Security*, vol. 11, no. 1, pp. 35–45, Sep. 2015.
[18] C. Jin, X. Feng, and Q. Shen, "Fully secure hidden ciphertext policy attribute-based encryption with short ciphertext size," in *Proc. Secur. Commun. Netw.*, Nov. 2016, pp. 91–98.
[19] Q. Wang, L. Peng, H. Xiong, J. Sun, and Z. Qin, "Ciphertext-policy attribute-based encryption with delegated equality test in cloud computing," *IEEE Access*, vol. 6, pp. 760–771, 2017.
[20] P. Chaudhari, M. L. Das, and A. Mathuria, "On anonymous attribute based encryption," in *Information Systems Security—ICISS* (Lecture Notes in Computer Science) vol. 9478, S. Jajoda, C. Mazumdar, Eds. Cham: Springer, 2015, pp. 378–392.
[21] Y. Zhang and D. Zheng, "Anonymous attribute-based encryption with large universe and threshold access structures," in *Proc. IEEE Int. Conf. Comput. Sci. Eng.*, Jul. 2017, pp. 870–874.
[22] Y. Rao, "A secure and efficient ciphertext-policy attribute-based signcryption for personal health records sharing in cloud computing," *Future Gener. Comput. Syst.*, vol. 67, pp. 133–151, Feb. 2017.
[23] N. Gorasia, R. R. Srikanth, N. Doshi, and D. J. Rupareliya, "Improving security in multi authority attribute based encryption with fast decryption," *Procedia Comput. Sci.*, vol. 76, pp. 632–639, Mar. 2016.
[24] W. Wang and M. He, "CP-ABE with hidden policy from Waters efficient construction," *Int. J. Distr. Sens. Netw.*, vol. 12, no. 11, Jan. 2016. Art. no. 3257029.
[25] R. Ostrovsky, A. Sahai, and B. Waters, "Attribute-based encryption with non-monotonic access structures," in *Proc. 14th ACM Conf. Comput., Commun. Secur.*, Oct. 2007, pp. 195–203.
[26] Q. Li, J. Ma, J. Xiong, T. Zhang, and X. Liu, "Fully secure decentralized key-policy attribute-based encryption," in *Proc. 5th Int. Conf. Intell. Netw. Collaborative Syst.*, Washington, DC, USA, Sep. 2013, pp. 220–225.
[27] Y. F. Tseng and C. Fan, "Cryptanalysis on the anonymity of li's ciphertext-policy attribute-based encryption scheme," in *Security With Intelligent Computing and Big-data Services—SICBS* (Intell Systems Computer), vol. 733, S. L. Peng and S. J. Wang, Eds. Cham, Switzerland: Springer, 2018, pp. 98–104.
[28] 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.
[29] X. Li, D. Gu, Y. Ren, N. Ding, and K. Yuan, "Efficient ciphertext-policy attribute based encryption with hidden policy," in *Internet and Distributed Computing Systems—IDCS* (Lecture Notes in Computer Science), vol. 7646, Y. Xiang, Eds. Berlin, Germany: Springer, 2012, pp. 146–159.
[30] B. Xu, L. Xu, H. Cai, L. Jiang, Y. Luo, and Y. Gu, "The design of an m-health monitoring system based on a cloud computing platform," *Enterprise Inf. Syst.*, vol. 11, no. 1, p. 17, Jan. 2017.

[31] H. Qian, J. Li, Y. Zhang, and J. Han, "Privacy-preserving personal health record using multi-authority attribute-based encryption with revocation," *Int. J. Inf. Secur.*, vol. 14, no. 6, pp. 487–497, Nov. 2015.
[32] 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.

LEYOU ZHANG received the M.S. and Ph.D. degrees from Xidian University, in 2002 and 2009, respectively, where he is currently a Professor. His current research interests include cryptography, network security, cloud security, and computer security.

GONGCHENG HU received the B.S. degree in mathematics from Xinxiang University, China, in 2017. He is currently pursuing the M.S. degree in applied mathematics with Xidian University, China. His current interests include applied cryptography and cloud security.

YI MU received the Ph.D. degree from Australian National University, in 1994. He is currently a Professor with Fujian Normal University (FJNU). Prior to his appointment at FJNU, he was a Full Professor with the School of Computing and Information Technology, University of Wollongong. He was involved in the areas of quantum cryptography, quantum computers, atomic computations, and quantum optics. His current research interests include cryptography, network security, access control, and computer security.

FATEMEH REZAEIBAGHA received the bachelor's degree in information technology engineering from Azad University, Iran, in 2009, the M.S. degree in information security from LTU, Sweden, in 2013, and the Ph.D. degree in computer science from the University of Wollongong, Australia, in 2017, where she is currently a Research Associate with SMART Infrastructure. Her major research interests include cryptography, blockchain, and information security.