Outsourced ciphertext-policy attribute-based encryption with partial policy hidden

Jinxia Yu, Guanghui He, Xixi Yan, Yongli Tang and Rongxia Qin

Abstract
Attribute-based encryption is an efficient and flexible fine-grained access control scheme. However, how to realize the attribute privacy concerns in the access policy and optimize the heavy computing overhead have been not adequately addressed. First, in view of the open-access policies formulated by data owners in the cloud environment and the linear growth of bilinear pairing operations with the number of attributes in the decryption process, a verifiable outsourced attribute-based encryption with partial policy hidden scheme is proposed, in which the attribute name of access policy can be sent while attribute value involving sensitive information can be hidden, so nobody can infer information from the access policy. Second, the bilinear pairing operation and modular power operation are outsourced to the cloud server, then users only need to perform constant exponential operation to decrypt. In addition, the proposed scheme is based on the composite order bilinear group and satisfies full secure under the standard model. Finally, compared with other schemes in term of function and performance, it shows that this scheme is more efficient and suitable for resource-constrained mobile devices in outsourcing environment.

Keywords
Attribute-based encryption, partial policy hidden, outsourced decryption

Date received: 17 December 2019; accepted: 22 April 2020

Handling Editor: Iftikhar Ahmad

Introduction
With the rapid development of cloud computing, it is increasingly favored by all fields of the Internet because of its powerful computing resources and storage performance, thus lead to a new computing pattern: outsourcing computing. Under cloud computing environment, users with limited computing and storage resources can outsource some time-consuming computing to the cloud server in order to reduce the burden. Based on the pay-as-you-go service, users can enjoy various information services provided by the cloud. When the data owner (DO) uploads the encrypted data to the cloud, he also loses actual control of the data. However, the cloud server is not completely trusted, so how to protect the private information in the uploaded data has become a popular research. Because DO encrypts data under access policy, ciphertexts can be decrypted by anyone with a set of attributes that satisfies the access policy. In attribute-based encryption (ABE) schemes, access policy is embedded in ciphertext implicitly and outsourced to cloud service provider (CSP) together with the ciphertext in a cloud environment. Because access policies are publicly available, everyone can access policies that contain private information. For example, in an electronic medical system, a patient...
authorizes a cardiologist to access encrypted data through the access policy as {Department: Cardiology; Doctor: Alice}. Without decryption, anyone who gains access to the encrypted data can conclude that the patient might suffer from “heart disease.” If the content of the attribute value in the policy is not visible, that is, the policy is set as {Department: XXX; Doctor: XXX}, then the patient’s privacy can be guaranteed. Thus, the idea that a policy is not uploaded to the cloud with ciphertext is produced by Nishide et al. Subsequently, Li et al. proposed an efficient ABE scheme with partial hidden policy. The scheme has less decryption cost, but the public parameters, ciphertext, and attribute information related to the policy are easily obtained by arbitrary malicious users. Therefore, Yin et al. proposed a more efficient scheme to supplement the deficiency of the scheme proposed by Li et al. in the standard model. And it was successfully reduced to the Decisional Bilinear Diffie–Hellman (DBDH) assumption.

However, most of the above schemes are based on the construction of prime order group. Cui et al. proposed a construction based on the composite order group supporting the attribute value hidden, but it only achieves selective security. In the application scenario of the electronic medical system, Zhang et al. not only implement policy concealment but also render less computing cost and storage overhead during the decryption process. In addition, the scheme is full secure under the standard model. However, the bilinear pairing operation and modular power operation involved in the decryption process are still associated with the number of attributes. To decrease the complicate pairing operation, Zhang et al. proposed a semi-hiding ABE scheme with a constant pairing operation; however, modular power operation remains linearly related to the number of attributes. To solve communication delay in remote cloud service centers, Xiong et al. introduced an attribute-based broadcast encryption scheme with partial policy hidden, user revocation, and outsourced decryption in the edge computing environment. The scheme has more practical function, but it was proved to be only selective secure.

In addition to implement partial policy hidden, the proposed scheme transfers the decryption load from user’s local device to the cloud. Therefore, the proposed scheme can easily be executed in resource-constrained devices (like mobile phones or Internet of things (IoT) devices). Because local computing nodes in an IoT environment are limited in computation and energy. For example, the merging of distributed ledger technology and IoT system is also the embodiment of outsourcing application in F Shahid et al.’s scheme. Therefore, to satisfy the typical IoT requirements, the proposed scheme transfers decryption load from local terminal to the cloud.

Our contribution

In order to preserve users’ privacy, the modern ABE schemes offer a “hidden access structure.” These schemes divide the “access structure” into two parts, namely, “attribute name” and “attribute value,” such that, attribute name is publicized and attribute value remains confidential. However, many of the existing ABE schemes are either inefficient or offer just a “selective security.” In almost all of the existing schemes, the decryption cost increases linearly with “number of attributes.” This article proposes a “full secure” ABE scheme with an outsourced decryption. The proposed system model transfers the decryption load from users’ local device to the highly resourceful cloud nodes. The proposed scheme has the characteristics of partial policy hiding and flexible expression. With respect to attribute hiding, the attribute is divided into two parts: attribute name and attribute value. When encrypting the data, the attribute value is only used to compute the ciphertext and not sent to the cloud along with the ciphertext. In addition,

1. It has a flexible expression in access policy. The proposed scheme employs linear secret sharing scheme (LSSS) to support “AND,” “OR,” and “Threshold” structure.
2. It offers outsourced decryption, while at the same time, and simultaneously eliminates redundant ciphertext components. In the decryption process, the bilinear pairing operation and modular power operation are outsourced to the CSP which will execute to compute the computation instead of the users. Users only need to verify the calculation results returned by CSP and perform a constant exponential operation to recover the plaintext.
3. It is proven to be full secure against chosen plaintext attack (CPA). Some of the abovementioned schemes satisfy selective secure by adding the “Init” stage, that is, adversary commits the challenge object in advance; however, selective secure is weaker than full secure. The proposed scheme was proven to be full secure through dual system encryption technology.
4. It has advantages in terms of performance and function. From the comparison and experiment, it shows that our scheme is more effective in terms of function, communication cost, and computing cost than the previously established schemes.

Related work

**ABE**

Sahai and Waters proposed a new public key cryptosystem, ABE, which can be divided into two categories
According to the location of the access policy: KP-ABE\textsuperscript{13} and CP-ABE\textsuperscript{14}, the policy in the KP-ABE scheme is embedded in the key and the policy in the CP-ABE scheme is contained in the ciphertext. Bethencourt et al.\textsuperscript{14} gave the CP-ABE scheme first, but only supported the “AND” gate structure. In order to achieve better ability of policy expression, Waters\textsuperscript{15} proposed the CP-ABE based on LSSS under the standard model. Since the policy is embedded in the ciphertext for CP-ABE, it means that the DO can set the policy to determine which attributes can access to the ciphertext, namely, making a fine-grained access control of the data. Compared to KP-ABE, CP-ABE is more widely used than KP-ABE, such as personal health care systems and fine-grained sharing in cloud storage.

**Policy hidden**

In order to resolve the privacy issue of user attributes in the cloud environment, Nishide et al.\textsuperscript{2} first proposed a CP-ABE scheme that supports the hiding of the access structure. The access structure is set implicitly in the ciphertext, which is not sent along with the ciphertext. Lewko et al.\textsuperscript{16} proposed a full secure CP-ABE scheme using the dual system encryption technology under the standard model. In Lai et al.’s\textsuperscript{3} scheme, the attribute is divided into two parts: attribute name and attribute value. The attribute name can be publicized, while the attribute value is hidden. And Jin et al.\textsuperscript{17} established a CP-ABE scheme supporting partial policy hiding. The ABE scheme achieves full secure, but the access structure uses the “AND” gate structure. In order to reduce the bilinear pairwise operation and modular exponentiation involved in decryption process, other specific schemes\textsuperscript{3,18–20} have been established. First, it is judged whether the attribute of users is matched with access policy before decryption first, if matched successfully, then the decryption operation is performed. But some of the proposed schemes\textsuperscript{18–20} only support “AND” gate structure, and the linear pair operation and modular exponentiation are still linearly related to the number of attributes during decryption process. Moreover, the schemes are proved to be only selective secure. Lai et al.\textsuperscript{3} adopted more flexible LSSS structure, and the scheme is full secure under the standard model. But the bilinear pairing operations and modular exponentiations involved in the user testing phase and decryption phase increase linearly with the complexity of the policy. Yan et al.\textsuperscript{21} introduced a multi-authority ABE of partial policy hidden with dynamic policy updating. In the above schemes, the policy hiding is only to hide the attribute value, hence it is called semi-hidden policy. Hao et al.\textsuperscript{22} used the Bloom filter to achieve complete hidden, that is, neither the attribute name nor attribute value is disclosed. In addition to judging whether the attribute information is in the set, the Bloom filter can also remove the mapping function between the matrix and the attribute, making the attribute completely hidden. However, the formal implementation of attribute privacy protection is not provided in the security proof process. In addition, the function of attribute hidden completely can also be realized using the inner product predicate encryption technology.\textsuperscript{23} Nevertheless, most of them only support the “AND” gate structure with weak expression ability, which limits in the actual application process.

**Outsourced ABE**

For the first time, Green et al.\textsuperscript{24} proposed an outsourced ABE scheme that is secure in random oracle model. The scheme commits the decryption operation to the decryption server provider (DSP), thus the ciphertext is converted into the Elgamal type, and then delivered to users to reduce the computing cost of data user (DU). Li et al.\textsuperscript{25} presented an offline/online ABE scheme, which reduces the computational overhead during the encryption phase using offline/online technology. Furthermore, “chameleon” hash function is introduced to implement verification before the decryption phase. The scheme is proved to satisfy the adaptive chosen ciphertext attack security, but the bilinear pairing operation involved in decryption process remains a large overhead for the user. Fan et al.\textsuperscript{26} introduced a verifiable outsourcing scheme for multi-authorization in cloud-fog computing, which outsources encryption and decryption to fog nodes close to the end user. Relative to the remote CSP, fog nodes can handle data with low latency, which is ideal choice for real-time calculation of data. Zhang et al.\textsuperscript{27} proposed an access control of full outsourcing scheme for the first time, in which the key generation, encryption, and decryption operations are all handled by the cloud, but it lacks a verification mechanism. Zhao et al.\textsuperscript{28} proposed a verifiable full outsourcing scheme based on the original scheme.\textsuperscript{27} The scheme supports verification and optimizes performance such that the computational cost does not increase significantly with the number of attributes or access policy complexity. For the issue of key leakage, Yan et al.\textsuperscript{29} proposed a traceable CP-ABE scheme in a multi-domain environment.

**Organization**

The rest of the article is organized as follows. Some background knowledges are provided in section “Preliminaries.” The algorithm definition and security model of outsourced CP-ABE are introduced in section “System architecture and security model.” Our outsourced CP-ABE of supporting partial policy hidden is stated in section “Our construction.” The scheme’s
security is analyzed in “Security proof.” The performance comparison is provided in “Performance analysis.” The article is concluded in section “Conclusion.”

Preliminaries

Composite order bilinear group

The scheme of this article is based on the design of composite order bilinear group whose order is the product of four distinct primes. Let \( \Phi \) be an algorithm that inputs security parameter \( 1^\lambda \) and outputs a tuple \( (p_1, p_2, p_3, p_4, G, G_T, e) \), where \( p_1, p_2, p_3, \) and \( p_4 \) are the four distinct primes, \( G \) and \( G_T \) are the cycle groups of order \( N = p_1 p_2 p_3 p_4 \), and \( e : G \times G \to G_T \) is a map function such that

1. **Bilinear:** \( \forall g, h \in G, a, b \in \mathbb{Z}_N, e(g^a, h^b) = e(g, h)^{ab} \).
2. **Non-degenerate:** if \( \exists g \in G \) such that the order of \( e(g, g) \) is \( N \) in \( G_T \).

Assuming that there is an group operation in \( G \) and \( G_T \) and the mapping function \( e \), it is computable in polynomial time in \( \lambda \). Let \( G_{p_1}, G_{p_2}, G_{p_3}, \) and \( G_{p_4} \) represent the subgroups of \( G \), the subgroups have order \( p_1, p_2, p_3, \) and \( p_4 \), respectively, then \( G = G_{p_1} \times G_{p_2} \times G_{p_3} \times G_{p_4} \). If \( g_1 \in G_{p_1}, g_2 \in G_{p_2} \), then \( e(g_1, g_2) = 1 \). If the elements in the mapping function \( e \) are elements of different subgroups, the equation still hold; thus, the composite order bilinear group is said to satisfy its orthogonality.

LSSS

The secret sharing scheme on the participant set \( P \) is called the LSSS if the following conditions are met.

1. A vector can be formed by the secret share of each party over \( \mathbb{Z}_{p_i} \).
2. For the secret sharing scheme \( \Pi \), there is a matrix \( M \) of size \( \ell \times n \) that maps each row of the matrix to an associated participant \( P \). For \( i = 1, \ldots, \ell \), \( \rho(i) \) is the party associated with the \( i \)th row of \( M \). We first generate a column vector \( v = (s, y_2, y_3, \ldots, y_\ell) \), where \( s \in \mathbb{Z}_{p_1} \) is a shared secret and \( n_1 \) is randomly selected, \( i = 2, \ldots, n \). According to scheme \( \Pi \), \( Mv \) is \( \ell \) secret shares of the shared secret \( s \), which indicates that \( \lambda_i = (Mv) \), held the secret share by the participants \( \rho(i) \).

The LSSS has the characteristics of linear reconstruction. If \( S \in A \) is an access authorization set, then there is a constant \( \{\omega_i \in \mathbb{Z}_{p_i}\}_{i \in I} \) that let \( \sum_{i \in I} \omega_i \lambda_i = s \) hold, where \( \lambda_i \) is the effective share of the secret \( s \), \( I = \{i : \rho(i) \in S\} \).

Complexity assumption

The security of this article is based on the following complexity assumptions, and a detailed description of the complexity assumptions is given.

**Assumption 1.** Given a group generation algorithm \( \Phi \), define the following distribution

\[
(N = p_1 p_2 p_3 p_4, G, G_T, e) \xrightarrow{R} \Phi, g \xrightarrow{R} G_{p_1},
X_1 \xrightarrow{R} G_{p_1}, X_4 \xrightarrow{R} G_{p_4}, D = (N, G, G_T, e, g, X_3, X_4),
T_1 \xrightarrow{R} G_{p_1} \times G_{p_2}, T_2 \xrightarrow{R} G_{p_3}
\]

If the advantage of adversary \( A \) satisfies \( \text{Adv}_{1\Phi, A}(\lambda) = |\Pr[A(D, T_1) = 1] - \Pr[A(D, T_2) = 1]| \), then this assumption can be broken.

**Definition 1.** For any probability polynomial time, if \( \text{Adv}_{1\Phi, A}(\lambda) \) is a negligible function, then the algorithm meets Assumption 1.

**Assumption 2.** Given a group generation algorithm \( \Phi \), define the following distribution

\[
(N = p_1 p_2 p_3 p_4, G, G_T, e) \xrightarrow{R} \Phi, g X_1 \xrightarrow{R} G_{p_1},
X_2, Y_2 \xrightarrow{R} G_{p_2}, X_3, Y_4 \xrightarrow{R} G_{p_4}, D = (N, G, G_T, e, g, X_1 X_2, Y_2 Y_3, X_3, X_4),
T_1 \xrightarrow{R} G_{p_1} \times G_{p_2} \times G_{p_3}, T_2 \xrightarrow{R} G_{p_1} \times G_{p_3}
\]

If the advantage of adversary \( A \) satisfies \( \text{Adv}_{2\Phi, A}(\lambda) = |\Pr[A(D, T_1) = 1] - \Pr[A(D, T_2) = 1]| \), then this assumption can be broken.

**Definition 2.** For any probability polynomial time, if \( \text{Adv}_{2\Phi, A}(\lambda) \) is a negligible function, then the algorithm meets Assumption 2.

**Assumption 3.** Given a group generation algorithm \( \Phi \), define the following distribution

\[
(N = p_1 p_2 p_3 p_4, G, G_T, e) \xrightarrow{R} \Phi, \alpha, s \xrightarrow{R} \mathbb{Z}_{N}, g \xrightarrow{R} G_{p_1},
g_2, X_2, Y_2 \xrightarrow{R} G_{p_2}, X_3 \xrightarrow{R} G_{p_3}, X_4 \xrightarrow{R} G_{p_4}, D = (N, G, G_T, e, g, g^\alpha X_2, g^\alpha Y_2, X_3, X_4),
T_1 = e(g, g)^{aT}, T_2 \xrightarrow{R} G_T
\]

If the advantage of adversary \( A \) satisfies \( \text{Adv}_{3\Phi, A}(\lambda) = |\Pr[A(D, T_1) = 1] - \Pr[A(D, T_2) = 1]| \), then this assumption can be broken.
Definition 3. For any probability polynomial time, if \( \text{Adv}_{\Phi, A}(\lambda) \) is a negligible function, then the algorithm meets Assumption 3.

Assumption 4. Given a group generation algorithm \( \Phi \), define the following distribution

\[
(N = p_1p_2p_3p_4, G, G_T, e) \overset{R}{\leftarrow} \Phi, \ell', \\
\ell' \overset{R}{\leftarrow} \mathbb{Z}_N, g, h \overset{R}{\leftarrow} G_{p_1}, g_2, X_2, A_2, B_2, D_2 \overset{R}{\leftarrow} G_{p_2}, \\
X_3 \overset{R}{\leftarrow} G_{p_3}, X_4, Z, A_3, D_4 \overset{R}{\leftarrow} G_{p_4}, \\
D = (N, G, G_T, e, g, g_2, g'_{B_2}, h'Y_2, X_3, X_4, hZ, g'_{D_2D_4}), \\
T_1 = h' A_2A_4, T_2 \overset{R}{\leftarrow} G_{p_1} \times G_{p_2} \times G_{p_4}
\]

If the advantage of adversary \( A \) satisfies

\[
\text{Adv}_{\Phi, A}(\lambda) = |\Pr[A(D, T_1) = 1] - \Pr[A(D, T_2) = 1]|,
\]

then this assumption can be broken.

Definition 4. For any probability polynomial time, if \( \text{Adv}_{\Phi, A}(\lambda) \) is a negligible function, then the algorithm meets Assumption 4.

System architecture and security model

System model

The system model proposed in this article is shown in Figure 1, including DO, storage service provider (SSP), DSP, DU, and attribute authority (AA).

The DO specifies an access policy and performs an encryption algorithm on the data using the public parameter \( (PK) \) generated by AA, which generates a complete ciphertext \( (CT) \), and outsources the policy of the hiding of attribute value along with the ciphertext \( (CT) \) to the cloud server.

The DU can access the ciphertext file in the cloud server and download the ciphertext from the SSP. Due to the terminal device of limited resources, the module exponent and the bilinear pairing operation involved in the decryption process are performed by the DSP that uses the transformed key \( (TK) \) generated by the AA. The DSP generates the transformed ciphertext and sends it to the DU. The DU verifies the computing result returned by the DSP and decrypts it with the secret key \( (SK) \) saved by itself and recovers the plaintext.

Algorithm definition

This article assumes that collusion does not occur between servers. AA is a key distribution center and is completely trusted. The algorithm contained in the scheme consists of the following six algorithms.

- \( \text{Setup}(\lambda) \rightarrow (PK, MSK) \): Create the public parameter \( (PK) \) and the master secret key \( (MSK) \) upon inputting security parameter \( \lambda \).

- \( \text{KeyGen}(PK, MSK, s) \rightarrow SK' \): The algorithm returns the secret key \( SK' \) by inputting the public parameter \( (PK) \), the master secret key \( (MSK) \), and the attribute set \( s \).

- \( \text{Encrypt}(PK, m, (M, \rho, \psi)) \rightarrow CT \): Generate a ciphertext \( (CT) \) upon receiving public parameter \( (PK) \), message \( m \), and access control policy \( (M, \rho, \psi) \) as input, where \( \psi \) is the set of attribute value that is sent along with \( CT \) to CSP.

- \( \text{KeyGen}_{out}(SK', z) \rightarrow (TK, SK) \): The algorithm outputs the transformed key \( (TK) \) for outsourcing decryption and the secret key \( (SK) \) of decryption by inputting \( SK' \) and selecting a random value \( z \in \mathbb{Z}_N \).

- \( \text{Transform}_{out}(TK, CT) \rightarrow CT' \): Generate partial decryption ciphertext \( CT' \) by inputting the transformed key \( (TK) \) and the ciphertext \( (CT) \).

- \( \text{Decrypt}(SK, CT') \rightarrow m \): The plaintext \( m \) is recovered by inputting the secret key \( (SK) \) of user’s local decryption and partially decryption ciphertext \( CT' \).

Security model

We define the security model of this article through the security game between simulator \( B \) and adversary \( A \). The game process is as follows:

- **Setup:** Challenger \( B \) performs the Setup algorithm and outputs the public parameter \( (PK) \) and the master secret key \( (MSK) \) to the adversary \( A \).

- **Phase 1:** The adversary \( A \) adaptively issues a polynomial bounded key query to the key generation oracle, query as follows:
O_{KeyGen}: The adversary submits the attribute set $s$, and the simulator executes the $KeyGen_{out}$ algorithm to output the secret key $SK_s$, which will be transmitted to adversary $A$.

**Challenge:** Adversary $A$ submits to simulator $B$ two equal length messages $m_0$ and $m_1$ and two access policies $A'_0 = (M, \rho, \psi_h)$ and $A'_1 = (M, \rho, \psi_f)$ with the restriction that none of them can be satisfied by any of the queried attribute sets in phase 1. $B$ flips a random coin $\beta \in \{0, 1\}$ and outputs the challenge ciphertext $CT_B$ under the access policy $A = (M, \rho, \psi_B)$ to $A$.

**Phase 2:** Similar to **Phase 1**, but adversary $A$ cannot query $B$ for a set of attributes that satisfy the policies $A'_0$ and $A'_1$.

**Guess:** The adversary $A$ outputs the guess value $\beta' = \{0, 1\}$, and if $\beta' = \beta$, then the adversary $A$ wins the game.

### Our construction

The verifiable outsourced ABE scheme proposed that supports policy hiding in the cloud environment is based on the scheme of Zhang et al., which can realize the attribute hiding in the policy and support the outsourced decryption process. The scheme is composed of the following seven algorithms, and the detailed description of each algorithm is as follows:

1. **Initialization:** $AA$ performs group generation algorithm $\Phi'(1^k)$, receives a security parameter $\lambda$ as input, outputs $(N, p_1, p_2, p_3, p_4, G, G_T, e)$, and sets the attribute universe $U = \mathbb{Z}_N$.

2. **Setup ($1^k$):** $AA$ selects $a, g, h \in \mathbb{G}_{p_1}$, $X_1 \in \mathbb{G}_{p_2}$, $Z, X_2 \in \mathbb{G}_{p_3}$, computes $H = h \cdot Z$, sets two hash functions $H_1 : \{0, 1\}^* \rightarrow \mathbb{Z}_N$, $H_2 : \{0, 1\}^* \rightarrow \{0, 1\}^\lambda$, and finally outputs the system $PK = (N, g, g^a, e(g, g)^{a}, (H, X_4, H_1, H_2)$ and the master secret key $MSK = (a, h, X_5)$.

   The user attribute set is defined as $s = \{x_5, S\}$, where $X_5$ represents the attribute name index, $X_5 \subseteq \mathbb{Z}_N$, and $S = \{s_i\}_{i \in X_5}$ represents the attribute value set.

3. **KeyGen ($PK, MSK, s$):** The algorithm chooses $t \in \mathbb{Z}_N$ and $R, R', R_i \in \mathbb{G}_{p_2}$, where $i \in X_5$, then returns $SK' = (K' = g^{ot} \cdot g^{at}R, L' = g^{\rho R'}, \{K' = (g^{ot}h') \cdot R_i\}_{i \in X_5})$.

4. **Encrypt ($PK, m, A$):** The DO specifies the access policy $A = (M, \rho, \Psi)$, where $M$ is a matrix of $\ell \times n$, and $\rho$ is a function that maps the $x$th row of the matrix $M$ to the attribute name index. And $\Psi = (l(x_1), l(x_2), \ldots, l(x_\ell) \in \mathbb{Z}_N$ is the attribute value set associated with access policy $(M, \rho)$.

   The algorithm selects $r \in G_T$ and computes $s = H_1(r, m)$ and $F = H_2(r)$. Taking the system public parameter ($PK$) and message $m$, the DO can receive $C = r \cdot e(g, g)^{a}$, $C' = g^{\rho}C = m \oplus R$. Then, select a vector $v = (s, v_1, v_2, \ldots, v_\ell)$, $r_s \in \mathbb{G}_{N}, Z_{d,x}$, and $Z_{d,x} \in \mathbb{G}_{p_2}$, compute $C_{1,x} = g^{\rho M \cdot v} \cdot (g^{\rho}H)^{r_s} \cdot Z_{c,x}$, $D_{1,x} = g^{\rho} \cdot Z_{d,x}$, and finally generate ciphertext $CT = ((M, \rho), C, C', C''_0, \{C_{1,x}, D_1 \}_{1 \leq x \leq \ell})$.

5. **KeyGen_{out}($SK', s$):** The algorithm inputs $SK' = (K' = g^{ot} \cdot g^{at}R, L' = g^{\rho R'}, \{K' = (g^{ot}h') \cdot R_i\}_{i \in X_5})$ generated by running the KeyGen algorithm then selects $z \in \mathbb{G}_{N}$ to obtain transformed key $TK = (K = K'^{1/z} = g^{ot/z} \cdot g^{at/z}R'^{1/z}, L = L'^{1/z} = g^{rt/z}R'^{1/z}, \{K_0 = K'_{1/z} = (g^{ot}h') \cdot R_i^{1/z}\}_{i \in X_5}$ and user’s secret key $SK = (z, TK)$.

6. **Transform_{out}($TK, CT$):** After receiving the transformed key ($TK$) and the ciphertext ($CT$), if the attribute set $s$ satisfies access policy $\rho$, there is a subset $\chi \in I_{(M, \rho)}$ that satisfies $\{l(\rho)i \in \chi \} \subseteq \chi_5$, where $I_{(M, \rho)} \subseteq \{1, 2, \ldots, \ell\}$ denotes the subset of $\{1, 2, \ldots, \ell\}$ that meets $(M, \rho)$, then there exists a set of constants $\{a_i\}_{i \in \chi}$ such that $\sum_{i \in \chi} a_i x_i = s$ holds, $x_\ell$ is the valid share of the secret $s$. The procedure of transformed ciphertext $CT_{\text{transform}}$ follows the steps below.

\[
CT_{\text{transform}} = e(C', K) = \prod_{i \in \chi} e(C_{1,i}, L') \cdot e(D_{1,i}, K_{0,i})^{a_i} = e(g, g)^{a/s}
\]

Finally, gain partial ciphertext $CT'' = (C, C', CT_{\text{transform}})$.

7. **Decrypt($SK, CT''$):** The algorithm is run by DU, which takes $CT'' = (C, C', CT_{\text{transform}})$ and $SK$ as input, then computes $r = C / CT''_{\text{transform}}$, $m = C \oplus H_1(r)$ and $s = H_1(r, m)$. If $C = r \cdot e(g, g)^{a}$, $CT_{\text{transform}} = e(g, g)^{a/s}$, then it indicates that the result returned by the cloud server is correct, and the plaintext $m$ is output finally.

### Security proof

**Theorem 1.** If the Assumptions 1–4 hold, then the proposed scheme based on the defined security model is proved to be full secure and satisfy CPA security.

The security proof of the scheme is similar to the literature, that is, the dual system encryption technology is used to prove its security. First, define two semi-functional structures: semi-functional ciphertext and semi-functional key. The normal secret key can decrypt normal ciphertext and semi-functional ciphertext, but
the semi-functional secret key cannot decrypt semi-functional ciphertext. And semi-functional key and semi-functional ciphertext are only used in security proof and do not appear in actual systems.

Semi-functional ciphertext: We set the CT_{normal} = (\text{GM}, p, C', \{C_{1,x}, D_{1,x}\}, 1 \leq x \leq l) as the normal ciphertext and select the generator g_2 of the subgroup G_{p_x}, C' \in \mathbb{Z}^n_{\mathbb{Z}} \times \mathbb{Z}^n_{\mathbb{Z}} and z_i \in \mathbb{Z}^n_{\mathbb{Z}} associated with the attribute, \( y_\text{e} \in \mathbb{Z}^n_{\mathbb{Z}} \) related to the number of matrix rows M_x. We can set the semi-functional ciphertext CT_{semi} = (C, C', \{C_{1,x}, D_{1,x}\}, 1 \leq x \leq l), where

\[
C = r \cdot e(\text{g}, g)^{y_\text{e}}, C' = g^2 \cdot g_2^z \\
C_{1,x} = g^{K_{1,x}}, (g^{e_i}(H)^{-s_{C_{1,x}}} \cdot Z_{c_{1,x}}, g_2 M_{i, a_i} + y_i z_0) \\
D_{1,x} = g^{K_{1,x}} \cdot Z_{d_{1,x}} \cdot g_2^{-z_{y_i}}
\]

Semi-functional key: we set SK_{normal} = (z, PK, K = g^{z_0} \cdot g_2^R, L = g^{z_0} R_0, \{K_i = (g^{h_i} h_i^z \cdot R_i^1, z_0)\}_{i \in \text{SK}}) and select \( d_1, d_2, (d_i)_{i \in \text{SK}} \in \mathbb{Z}^n_{\mathbb{Z}} \). The semi-functional key may be one of the following three cases:

Semi-functional key of type 1

\[
K_1 = g^{z_0} \cdot g_2^R \cdot g_2^d, L = g^{z_0} R_0^1 \cdot g_2^d, \{K_i = (g^{h_i} h_i^z \cdot R_i^{1, z_0})\}_{i \in \text{SK}}
\]

Semi-functional key of type 2

\[
K_1 = g^{z_0} \cdot g_2^R \cdot g_2^{z_0}, L = g^{z_0} R_0^1 \cdot g_2^d, \{K_i = (g^{h_i} h_i^z \cdot R_i^{1, z_0})\}_{i \in \text{SK}}
\]

Semi-functional key of type 3

\[
K_1 = g^{z_0} \cdot g_2^R \cdot g_2^{z_0}, L = g^{z_0} R_0^1 \cdot g_2^d, \{K_i = (g^{h_i} h_i^z \cdot R_i^{1, z_0})\}_{i \in \text{SK}}
\]

The proof of this article’s security is based on the Assumptions 1–4 and is demonstrated through a series of games, now we define the following games:

Game_{k} := In this game, we challenge ciphertext is semi-functional, the first k keys are semi-functional of type 3 and the latter keys are normal. Note that the challenge ciphertext in the Game_{k,3} is semi-functional, and all keys are semi-functional of type 3.

Game_{F0} := In this game, the challenge ciphertext is a semi-functional ciphertext generated by selecting an encrypted message from m_0 and m_1 given the adversary, and all keys are semi-functional of type 3.

Game_{F1} := The last game is the same as Game_{F0}, except that C_{1,x} in the challenge ciphertext is the random elements in the group G_{p_x} \times G_{p_x} \times G_{p_x}. However, the challenge ciphertext in the game is one of the opponents randomly selected in \([\psi_0, \psi_1]\), so the advantage of the adversary A is 0.

We can prove that these games are indistinguishable based on Lemmas 1–6. If the Assumptions 1–4 hold, it can be proved that Game_{real} and Game_{e3} are indistinguishable, so the adversary does not exist to break this solution with a non-negligible advantage.

Lemma 1: Suppose the algorithm \( \Phi \) satisfies Assumption 1, then the Game_{real} and Game_{e0} are computationally indistinguishable.

Proof: If there exists an advantage of adversary \( A \) such that \( \text{Adv}_{\text{real}, \text{real}} - \text{Adv}_{\text{e0}, \text{e0}} = \varepsilon \) in Game_{real} and Game_{e0}, then we can construct a simulator B to break Assumption 1 with \( \text{Adv}_{\text{real}, \text{real}}(A) = \varepsilon. B \) gets \( g, X_{1}, X_{2}, T \) and simulates Game_{real} or Game_{e0} with adversary A.

Setup: B chooses \( \alpha, a_0, a_1, Z \in G_{p_x} \), then sets \( Y = e(g, g)^a, h = g^{a_0}, H = h \cdot Z \) and sends the system public parameter (PK) to A.

Phase 1: B has the response of A’s key request. Since B knows the master secret key MSK = (\alpha, h, X_{1}), it can generate a normal key through the key generation algorithm.

Challenge: Adversary A submits to simulator B two equal length messages \( m_0 \) and \( m_1 \) and two access policies \( A' = (M, \rho, \psi_0) \) and \( A' = (M, \rho, \psi_1) \) with the restriction that none of them can be satisfied by any of the queried attribute sets in phase 1. Let \( \psi_{\beta} = (t_{\beta(1)}, t_{\beta(2)}, \ldots, t_{\beta(l)}) \). B flips a random coin \( \beta \in \{0, 1\} \) and performs the following steps:

1. B creates \( \tilde{v} = (\tilde{v}_1, \tilde{v}_2, \ldots, \tilde{v}_n) \), where \( \tilde{v}_i \in \mathbb{Z}^n_{\mathbb{Z}} \) and \( 2 \leq i \leq n \).
2. B selects \( \tilde{v}_x \in \mathbb{Z}^n_{\mathbb{Z}}, \tilde{z}_x = (a_0 + t_{\beta(x)})^{-1}, \tilde{Z}_{c_{1,x}}, Z_{d_{1,x}} \in G_{p_x} \), where \( 1 \leq x \leq \ell \).
3. B selects \( \tilde{s} \in \mathbb{Z}^n_{\mathbb{Z}}, \psi_{\beta} = (t_{\beta(1)}, t_{\beta(2)}, \ldots, t_{\beta(l)}) \) and obtains \( m_{\beta} \) associated the \( m_{\beta} \), then it does the following calculations

\[
C = r_B \cdot e(g^{a_0}, T), C' = T, C'' = m_{\beta} \oplus F \\
C_{1,x} = T^{a_{\beta,x}} \cdot T^{-(a_0 + i_{\beta(x)})} \tilde{Z}_{c_{1,x}}, D_{1,x} = T^{z_{R_{1,x}}}
\]
4. $B$ sets the challenge ciphertext as $CT_B = ((M, \rho), C', C', \{C_1, x, D_1, 1\}_{1 \leq i \leq l})$ and sends it to $A$. If $T \leftarrow G_{p_1} \times G_{p_2}$, let $T = g^r g_2^s$, and
\[
C = r B (g, g)^{\alpha}, \quad C' = g^r g_2^s \\
C_{1, x} = T^{M - a_0 + t_{p(x)}} Z_{c, x} g_2^{M - w} + \gamma_{x}, z_{p(x)} \\
D_{1, x} = T^{r} Z_{d, x} g_2^r 
\]

where $\gamma_{x} = - c \gamma_{x}$

Hence, this challenge ciphertext is semi-functional and $B$ simulates game $Game_{0}$ if $T \leftarrow G_{p_1}$, then $B$ simulates $Game_{real}$ and it is normal ciphertext.

**Phase 2:** Similar to Phase 1, but adversary $A$ cannot query $B$ for a set of attributes that satisfy the policies $A_0$ and $A_1$. Note that, if $T \leftarrow G_{p_1} \times G_{p_2}$, then $B$ simulates $Game_{0}$; if $T \leftarrow G_{p_1}$, then $B$ simulates $Game_{real}$. Finally, $B$ can distinguish $T$ and $Adv1_{\Phi, A}(\lambda) = \nu$ according to the output of $A$.

**Lemma 2:** Suppose the algorithm $\Phi$ satisfies Assumption 2, then the $Game_{k, 1}$ and $Game_{k, 1}$ are computationally indistinguishable.

**Proof:** If there exists an adversary of adversary $A$ such that $|Game_{k, 1, 1} | Adv_{A} - |Game_{k, 1} | Adv_{A} | = \nu$ in $Game_{k, 1}$ and $Game_{k, 1}$, then we can construct a simulator $B$ to break Assumption 2 with $Adv1_{\Phi, A}(\lambda) = \nu$. $B$ gets $g, X_1, X_2, Y_1, Y_2, X_3, X_4$, and $T$ and simulates $Game_{k, 1}$ or $Game_{k, 1}$ with adversary $A$.

**Setup:** $B$ chooses $a, a_0, a_0 \in g_{Z, N}$, and $Z \in g_{G_{p_1}}$, then sets $\gamma_{r} = \epsilon(g, g)^{\alpha}, h = \epsilon^{\alpha}$, and $H = h \cdot Z$ and sends the system public parameter (PK) to $A$. $B$ has the response of $A$’s key query. Since $B$ knows the master secret key, it can generate a normal key through the key generation algorithm.

**Phase 1:** $B$ answers the $j$th secret key query associated with attribute set $s = (\chi, S)$. The procedure is as follow:

1. For $j < k$, $B$ selects $t, d, d' \in g_{Z, N}$, $\{d_i \in g_{Z, N}\}_{1 \leq i \leq l}$, and creates a semi-functional key of type 3. The specific steps are as follows
\[
K = g^{\alpha} g^{a_0} (Y_2 Y_3)^{\gamma}, L = g^{(Y_2 Y_3)^{\gamma}} \\
\{K_i = (g^h (Y_2 Y_3)^{\gamma}) \}_{1 \leq i \leq l}
\]

Note that it is semi-functional of type 3, because the value of $t, d, d' \in g_{Z, N}$ modulo $p_2$ is not uncorrelated to the value of the modulo $p_3$.

2. For $j > k$, because $B$ knows the master secret key $MSK = (\alpha, h, X_3)$, it can generate normal key through the key generation algorithm.

3. $B$ responds to the $k$th secret key query and randomly selects $R, R' \in g_{Z, N}$, where $i \in X_5$, then computes $K = g^{\alpha} T^{R} R', L = T^{R'}$, and $\{K_i = T^{a_0} + s R_i \}_{1 \leq i \leq l}$. The process is as follow:

(a) If $T \leftarrow G_{p_1} \times G_{p_2} \times G_{p_1}$, let $T = g^{r} g_2^s \hat{R}$, then generate semi-functional key of type 1
\[
K = g^{\alpha} g^{a_0} R g_2^s, \quad L = g^{r} R^d g_2^s \\
\{K_i = (g^h R g_2^{d'}) \}_{1 \leq i \leq l}
\]

where
\[
R = \hat{R}^d R, \quad d = a d', \quad R' = R \hat{R}^d \\
R_t = \hat{R}^{a_0} + s R_t, \quad z_i = a_0 + s_i
\]

Note that the value of $a, a_0, s_i$ modulo $p_2$ is not uncorrelated to the value of the modulo $p_3$.

(b) If $T \leftarrow G_{p_1} \times G_{p_2}$, it is normal key.

**Challenge:** Adversary $A$ submits to simulator $B$ two equal length messages $m_0$ and $m_1$ and two access policies $A_0 = (M, \rho, \psi_0)$, $A_1 = (M, \rho, \psi_1)$ with the restriction that none of them can be satisfied by any of the queried attribute sets in Phase 1. $B$ flips a random coin $\beta \in \{0, 1\}$ and performs the following steps:

1. $B$ creates $\bar{\nu} = (1, \bar{v}_2, \ldots, \bar{v}_n)$, where $\bar{v}_i \in g_{Z, N}$, $2 \leq i \leq n$.
2. $B$ selects $\bar{r}_i \in g_{Z, N}$, $\tilde{s}_i = (a_0 + t_{p(x)})^{-1}$, $\tilde{Z}_{c, x}$, and $Z_{d, x} \in g_{G_{p_1}}$, where $1 \leq x \leq l$.
3. $B$ selects $s \in g_{Z, N}$ and obtains $r_\beta$ associated the $m_\beta$, then it does the following calculations
\[
C = r_\beta \cdot g^{a_0} X_1 X_2, \quad C' = X_1 X_2, \quad C'' = m_\beta \oplus F \\
C_{1, x} = X_1 X_2 a_{M, t} \bar{v}_2 X_2^{a_0 + t_{p(x)}} Z_{c, x} \\
D_{1, x} = X_1 X_2 \tilde{Z}_{d, x}
\]

4. $B$ sets the challenge ciphertext as $CT_B = ((M, \rho), C, C', C', \{C_{1, x, D_1, x, 1} \}_{1 \leq i \leq l})$ and sends it to $A$. If $T \leftarrow G_{p_1} \times G_{p_2}$, let $T = g^{r} g_2^s$, then
\[
C = r B (g, g)^{\alpha}, \quad C' = g^{r} g_2^s \\
C_{1, x} = T^{M - a_0 + t_{p(x)}} Z_{c, x} g_2^{M - w} + \gamma_{x}, z_{p(x)} \\
D_{1, x} = T^{r} Z_{d, x} g_2^r 
\]

where
\[ v = s\tilde{v}, \quad v_1 = s, \quad r_x = s\tilde{r}_x, \quad Z_{c,x} = Z^{p}\tilde{Z}_{c,x} \]
\[ z_{p(x)} = a_0 + t_{p(x)}, \quad \omega = cav, \quad \gamma_x = -cr_x \]

We know this challenge ciphertext is semi-functional and the value of \( a, a_0, \{ t_{p(x)} \}_{x \in \mathbb{Z}, \gamma_x \sim \mathcal{N}} \) is not uncorrelated to the value of the modulo \( p_2 \).

**Phase 2:** Similar to Phase 1, but adversary \( A \) cannot query \( B \) for a set of attributes that satisfy the policies \( A_0 \) and \( A' \). Note that, if \( T \leftrightarrow G_{p_1} \times G_{p_2} \times G_{p_3} \), then \( B \) simulates \( \text{Game}_{k,1} \); if \( T \leftrightarrow G_{p_1} \times G_{p_3} \), then \( B \) simulates \( \text{Game}_{k,2} \). Finally, \( B \) can distinguish \( T \) and \( \text{Adv}_{2\Phi, A}(\lambda) = \varepsilon \) according to the output of \( A \).

**Lemma 3.** Suppose the algorithm \( \Phi \) satisfies Assumption 2, then the \( \text{Game}_{k,1} \) and \( \text{Game}_{k,2} \) are computationally indistinguishable.

**Proof.** If there exists an advantage of adversary \( A \) such that \(| \text{Game}_{k,1} - \text{Game}_{k,2} | = \varepsilon \) in \( \text{Game}_{k,1} \) and \( \text{Game}_{k,2} \), then we can construct a simulator \( B \) to break Assumption 2 with \( \text{Adv}_{2\Phi, A}(\lambda) = \varepsilon \). \( B \) gets \( x_1, x_2, y_1, y_2, x_3, x_4 \), and \( T \) and simulates \( \text{Game}_{k,2} \) or \( \text{Game}_{k,3} \) with adversary \( A \).

**Setup:** \( B \) chooses \( \alpha, \beta, a_0 \in \mathbb{Z}_N \), and \( Z \in \mathbb{G}_{p_3} \), then sets \( Y = e(g, g)^a, h = g^{a_0} \), and \( H = h \cdot Z \) and sends the system private parameter (PK) to \( A \). \( B \) has the response of \( A \)’s key request. Since \( B \) knows the master secret key \( MSK = (\alpha, h, X_3) \), it can generate a normal key through the key generation algorithm.

**Phase 1:** \( B \) answers the \( j \)th secret key query associated with attribute set \( s = (X_3, S) \). The first \( k - 1 \) type 3 semi-functional keys and the \( k \)th and the following are constructed as normal keys according to the method of Lemma 2.

Similarly, to answer the \( k \)th key request, \( B \) additionally selects \( \tau \in \mathbb{Z}_N \), and sets \( K = g^\tau T^{\tau}(Y_2 Y_3) \), \( L = T^{\tau} Y_2 \), and \( \{ K_i = T^{\tau + \tau_i} \}_{i \in \mathbb{Z}} \). The reason of making this change is to add \( (Y_2 Y_3) \) term which randomizes the \( G_{p_1} \) part of the key component \( K \). If \( T \leftrightarrow G_{p_1} \times G_{p_2} \times G_{p_3} \), it is semi-functional key of type 1; if \( T \leftrightarrow G_{p_1} \times G_{p_3} \), it is semi-functional key of type 2.

**Challenge:** Same as Lemma 2.

**Phase 2:** Similar to Phase 1, but adversary \( A \) cannot query \( B \) for a set of attributes that satisfy the policies \( A_0 \) and \( A' \). Note that, if \( T \leftrightarrow G_{p_1} \times G_{p_2} \times G_{p_3} \), then \( B \) simulates \( \text{Game}_{k,1} \); if \( T \leftrightarrow G_{p_1} \times G_{p_3} \), then \( B \) simulates \( \text{Game}_{k,2} \). Finally, \( B \) can distinguish \( T \) and \( \text{Adv}_{2\Phi, A}(\lambda) = \varepsilon \) according to the output of \( A \).

**Lemma 4.** Suppose the algorithm \( \Phi \) satisfies Assumption 2, then the \( \text{Game}_{k,2} \) and \( \text{Game}_{k,3} \) are computationally indistinguishable.

**Proof.** If there exists an advantage of adversary \( A \) such that \(| \text{Game}_{k,2} - \text{Game}_{k,3} | = \varepsilon \) in \( \text{Game}_{k,2} \) and \( \text{Game}_{k,3} \), then we can construct a simulator \( B \) to break Assumption 2 with \( \text{Adv}_{2\Phi, A}(\lambda) = \varepsilon \). \( B \) gets \( x_1, x_2, y_1, y_2, x_3, x_4 \), and \( T \) and simulates \( \text{Game}_{k,2} \) or \( \text{Game}_{k,3} \) with adversary \( A \).

**Setup:** \( B \) chooses \( \alpha, \beta, a_0 \in \mathbb{Z}_N, \) and \( Z \in \mathbb{G}_{p_3} \), then sets \( Y = e(g, g)^a, h = g^{a_0} \), and \( H = h \cdot Z \) and sends the system public parameter (PK) to \( A \). \( B \) has the response of \( A \)’s key request. Since \( B \) knows the master secret key \( MSK = (\alpha, h, X_3) \), it can generate a normal key through the key generation algorithm.

**Phase 1:** \( B \) answers the \( j \)th secret key query associated with attribute set \( s = (X_3, S) \). The first \( k - 1 \) type 3 semi-functional keys and the \( k \)th and the following are constructed as normal keys according to the method of Lemma 2.

\( B \) responds to the \( k \)th secret key query and randomly selects \( r, \tau \in \mathbb{Z}_N, R, R' \), and \( R_i \in \mathbb{Z}_{p_2} \), where \( i \in \chi_{X_3} \), computes \( K = g^\tau T^{\tau}(Y_2 Y_3) \), \( L = T^{\tau} R \), \( \{ K_i = T^{\tau + \tau_i} R_i \}_{i \in \mathbb{Z}} \). The procession is as follow:

(a) If \( T \leftrightarrow G_{p_1} \times G_{p_2} \times G_{p_3} \), let \( T = g^{r_1} g^{\tau} R \), then generate semi-functional key of type 3

\[ K = g^{a_0} T^{\tau}(Y_2 Y_3), \quad L = g^{\tau} R, \quad \{ K_i = (g^{a_0} h)^i R g^{\tau} \}_{i \in \chi_{X_3}} \]

(b) If \( T \leftrightarrow G_{p_1} \times G_{p_3} \), it is semi-functional key of type 2.
Then, semi-functional key of type 3 is created as queried attribute sets in Phase 1.

Assumption 3 with $A^1 = (M, \rho, \psi_0)$ and $A^1' = (M, \rho, \psi_1)$ with the restriction that none of them can be satisfied by any of the queried attribute sets in Phase 1. $B$ flips a random coin $\beta \in \{0, 1\}$ and performs the following steps:

1. $B$ creates $\tilde{v} = (1, \tilde{v}_2, \ldots, \tilde{v}_n)$, where $\tilde{v}_i \in \mathbb{Z}_N$, $2 \leq i \leq n$.
2. $B$ selects $\tilde{r}_x \in \mathbb{Z}_N$, $\tilde{Z}_{c,x}$, and $Z_{d,x} \in \mathbb{G}_{p^t}$, where $1 \leq x \leq \ell$.
3. $B$ selects $\tilde{s} \in \mathbb{Z}_N$, lets $\psi_B = (t_{p(1)}, t_{p(2)}, \ldots, t_{p(\ell)})$, and obtains $r_B$ associated the $m_B$, then it does the following calculations

$$
C = r_B \cdot T, \quad C' = g^g Y_2, \quad C'' = m_B \oplus F\
C_{1,x} = (g^g Y_2)^{a_0} \cdot \tilde{v}_i (g^g Y_2)^{-a_0} \cdot r_B \tilde{Z}_{c,x}\
D_{1,x} = (g^g Y_2)^{\tilde{v}_i} Z_{d,x}
$$

4. $B$ sets the challenge ciphertext as $CT_B = ((M, \rho), C, C', C'', \{C_{1,x}, D_{1,x}\})_{1 \leq x \leq \ell}$ and sends it to $A$. If $g^g Y_2 = g^g g_{2}^{t}$, then

$$
C = r_B (g^g Y_2)^{a_0}, \quad C' = g^g g_{2}^{t}\
C_{1,x} = T^{a_0} \cdot \tilde{v}_i (g^g Y_2)^{-a_0} \cdot r_B \tilde{Z}_{c,x}\nD_{1,x} = (g^g Y_2)^{\tilde{v}_i} Z_{d,x} = g^g Z_{d,x} g_{2}^{t - \gamma_x}
$$

where

$$
\nu = s, \quad v_1 = s, \quad r_x = s \tilde{r}_x, \quad Z_{c,x} = s \tilde{Z}_{c,x}\n\gamma_x = -ct_{p}\n$$

Note that the value of $a, a_0, \{t_{p(\ell)}\}_{1 \leq x \leq \ell}$, $s, \{s_i\}_{1 \leq x \leq \ell}$, $\tilde{r}_x$, and $\tilde{s}$, $\{\tilde{v}_i\}_{1 \leq x \leq \ell}$ modulo $p_1$ is not uncorrelated to its value modulo $p_2$.

**Phase 2**: Similar to Phase 1, but adversary $A$ cannot query $B$ for a set of attributes that satisfy the policies $A_0'$ and $A_1'$. Note that, if $T = \varepsilon (g^g Y_2)^{\gamma_x}$, the challenge ciphertext is a semi-functional ciphertext generated by encryption of $m_B$, then $B$ simulates $Game_{e_0}$, Otherwise, it is a semi-functional ciphertext that is encrypted by random messages in $G_{\ell}$, then $B$ simulates $Game_{e_0}$ or $Game_{F_1}$ with adversary $A$.

**Proof**: If there exists an advantage of adversary $A$ such that $|Game_{e_0} - Game_{F_1}| = \varepsilon$ in $Game_{e_0}$ and $Game_{F_1}$, then we can construct a simulator $B$ to break Assumption 4 with $Adv_3(\Phi, A) = \varepsilon$. $B$ gets $g, g_{2}, Y_1, Y_2, X_1, X_2, Y_3, X_3, X_4$, and $T$ and simulates $Game_{e_0}$ or $Game_{F_1}$.

**Lemma 6**: Suppose the algorithm $\Phi$ satisfies Assumption 4, then the $Game_{e_0}$ and $Game_{F_1}$ are computationally indistinguishable.
Table 1. Comparison of performance.

| Scheme          | Policy hiding | Full secure | Outsource decryption | Group order | Standard model |
|-----------------|---------------|-------------|----------------------|-------------|----------------|
| Lai et al.²      | ✓             | ✓           |                     | Composite   | ✓              |
| Yin et al.³      | ✓             | ✓           |                     | Prime       | ✓              |
| Zhang et al.⁷    | ✓             | ✓           |                     | Composite   | ✓              |
| This article     | ✓             | ✓           | ✓                   | Composite   | ✓              |

Table 2. Comparison of communication overhead.

| Scheme          | PK size | SK size | CT size |
|-----------------|---------|---------|---------|
|                 | G₀      | G₁      | G₀      | G₁      |
| Lai et al.²      | N + 4   | | S| + 2 | 4| + 2 | 2 |
| Zhang et al.⁷    | 4       | | S| + 2 | 3| + 2 | 2 |
| This article     | 4       | | S| + 3 | 2| + 1 | 1 |

PK: public parameter; SK: secret key; CT: ciphertext.

C = G₀, C' = g₀g₂, C'' = m₀ ⊕ F
C₁x = g₀M₀y(x)g₁D₂D₄⁻¹rx₁₀T⁻¹ry₁₀g₂M₀wZcₓ
D₁x = (g₀D₂D₄)₁₀

4. B sets the challenge ciphertext as CTₐ = ((M₀,p₀), C, C', C'', {C₁x, D₁x}₁≤x≤l) and sends it to A. If T = h₁A₂A₄, and suppose h = g₀, D₂ = g₀, A₂ = g₁t₀, T₁, T₂, γ ∈ G₀[N], then
C = G₀, C' = g₀g₂
C₁x = g₀M₀y(x)g₁D₂D₄⁻¹rx₁₀g₂M₀w + y₀γ₀Zcₓ
D₁x = (g₀D₂D₄)₁₀

where
rₓ = r₀⁻¹rx, Zcₓ = D₄⁻¹γ₀⁻¹A₄⁻¹Zcₓ, Z₀ = -c₀rx, Zdₓ = D₄⁻¹

Phase 2: Similar to Phase 1, but adversary A cannot query B for a set of attributes that satisfy the policies A₀ and A₁. Note that, if T = h₁A₂A₄, the challenge ciphertext is a semi-functional ciphertext generated by encryption of m₀, then B simulates Gameₚ₀. If T ← Tₚ₀ × Tₚ₀ × Tₚ₀, it is a semi-functional ciphertext that is encrypted by random messages in G₀, and C₁x is the random element in G₁ × G₁ × G₁, then B simulates Game₀₁. Finally, B can distinguish T and Advₐₙₐₜₐ(λ) = ε according to the output of A.

Performance analysis

The scheme of this article is compared with that of Lai et al.,³ Yin et al.,⁵ and Zhang et al.,⁷ from the perspectives of function and performance. In the scheme comparison process, Gₚ₀ represents the subgroup of the order p₀, and Gₚ₀ represents the subgroup of the order p₀p₁. N represents the number of attribute universe, |l| represents the number of the matrix M row, |S| represents the number of user attributes, and |y| represents the number of attribute sets that satisfy the policy. We use Exp, Expₜ, and P to denote a module exponential operation in G, a module exponential operation in G₀, and a bilinear pair operation, respectively. Because the main computing overhead of this scheme contains linear pairwise operation and modular exponentiation, module multiplication and hash operation can be ignored.

Theoretical analysis

Table 1 mainly compares the functionalities of the schemes. It can be seen that the schemes all implement the policy-hiding function, and they are all proven to be full secure apart from the scheme proposed by Yin et al.,³ is selective secure under the standard model and this scheme⁵ does not support outsourced decryption operations. However, the schemes of Lai et al.,³ and Zhang et al.,⁷ are constructed based on the composite order group, which realizes the full security of the scheme, but does not support outsourcing decryption as well. The proposed scheme is based on the composite order group, and it not only realizes policy hidden but also supports outsourced decryption operations, which reduces the user’s computational cost. In addition, it is proved to satisfy full secure under the standard model.

Table 2 mainly shows the comparison of the public key size (PK size), user’s secret key (SK size), and ciphertext size (CT size). Because the scheme proposed by Yin et al.,⁵ is based on the prime order group and
the rest of the schemes are the composite order group, only the schemes of Lai et al.,3 Zhang et al.,7 and this article are analyzed in Table 2. It can be seen from Table 2 that the size of the proposed scheme’s secret key is $|S| + 3$, while the schemes of Lai et al.3 and Zhang et al.7 are both $|S| + 2$. However, the $C^T$ size of the proposed scheme is much shorter by $|t| + 1$ in scheme of Zhang et al.7 and $2(|t| + 1)$ in scheme of Lai et al.3 and the $PK$ size of the proposed scheme is shorter by $N$ in scheme of Lai et al.3.

Table 3 shows the computational overhead. In the encryption process, the exponent operation involved in this scheme is $(4(|t| + 1)Exp + 1ExpT$, which is much lower than other schemes of Lai et al.3 and Zhang et al.7 but slightly higher than the scheme proposed by Yin et al.5 Since the outsourcing decryption operation is not supported in schemes of Lai et al.,3 Zhang et al.7 and Yin et al.,5 the decryption process can be separated into two phases: decryption match test and data decryption. It is first judged whether to match the policy in decryption match test period. If matched successfully, then perform the decryption operation. But the linear pair operation and the modular exponentiation operation during the decryption period are performed, which are still linearly increasing with the number of attributes. The scheme transfers the most decryption operation to the cloud server, and the cost of outsourced decryption is still less than other schemes, so the DU in this scheme only needs 1 exponent time to complete the decryption process, which greatly reduces the user’s computational overhead.

The proposed scheme makes a trade-off between ciphertext size and cost of decryption. More precisely, the scheme accepts unnecessary exercises of decryption process in order to reduce the ciphertext size. However, because decryption is performed by sufficiently powerful cloud nodes; therefore, additional decryption exercises are acceptable. From the analysis above, our scheme not only has the characteristic of outsourcing decryption and policy hidden but also has certain performance advantages.

### Experiment analysis

Through the above theoretical analysis, our scheme has more advantages in terms of function and efficiency. In order to more accurately evaluate the actual performance of the scheme, we further analysis computing overhead through experiments, including the encryption time and the decryption time.

**Experimental environment:** Windows 10, Inter® Core™ i5-8300H (2.30 GHz), memory 8 GB, the experimental code is based on java pairing-based cryptography library (JPBC-2.0.0), and MyEclipse development environment. In the experiment, the paired structure of type A is used to construct an elliptic curve $y^2 = x^3 + x$ on a finite field. The order of the group is $r$, and the order of the base field is $q$. Here, we take $r = 160$ bit, $q = 512$ bit, where the pairing operation and modular exponent invoked pairing.pairing() and G-1.powZn(), respectively, in the library for testing.

**Experimental setup:** In the ciphertext-policy ABE scheme, the number of attributes in the access policy affects the encryption and decryption time. In the experiment process, because the high computational overhead is outsourced to the cloud server for calculation, we only test the computing time of the terminal device. In the experiment, we set the number of attributes as 50 and increase by 10 the number of attributes each time, generating five different access policies. By comparing the computing time of the terminal user under different access policies, we tested 10 times for each policy and took an average value as the experimental result.

Figure 2 has two subgraphs (Figure 2(a) and (b)), which represents the execution time of the DO’s encryption and the execution time of the DU’s decryption. In Figure 2(a), we can see that the execution time of the proposed scheme is lower than the computational time of other schemes of Lai et al.3 and Yin et al.5 but slightly higher than the calculation time of the scheme proposed by Yin et al.5 In Figure 2(b), during the procedure of user’s decryption, the schemes of Lai et al.,3 Yin et al.,5 and Zhang et al.7 do not support the outsourced decryption operation. And they involve the

| Scheme          | Encryption phase | Cloud server | DU |
|-----------------|-----------------|--------------|----|
| Lai et al.3     | $(7|t| + 2)Exp + 2ExpT$ | –            | $1y[Exp + 1ExpT + (|y| + 2)P]$ |
| Yin et al.5     | $(|t| + 2)Exp + 3ExpT + 1P$ | –            | $1ExpT + 1P$ |
| Zhang et al.7   | $(6|t| + 2)Exp + 2ExpT$ | –            | $2y[Exp + 2P]$ |
| This article    | $(4|t| + 1)Exp + 1ExpT$ | $|y[ExpT + (2|y| + 1)P]$ | $1ExpT$ |

DO: data owner; DU: data user.
computational cost of the bilinear pair operation and the computational cost increasing linearly with the number of attributes. This scheme achieves outsourcing decryption operations and verification operations, so the user only needs to perform constant exponential operation, which greatly reduces the user’s computational overhead.

By comprehensive analysis, the proposed scheme is superior to other schemes in terms of function, communication, and computational burden, so it is effective and feasible in the application of cloud computing.

**Conclusion**

In order to solve the problem of privacy protection and heavy computing overhead in the cloud outsourcing environment, this article proposes a partial policy-hiding ABE scheme that can verify the result of outsourcing decryption. The privacy of user attributes is realized by dividing attributes into two parts: attribute name and attribute value, attribute name disclosure and attribute value concealment. For the bilinear pairing operation and module power operation, this article will decrypt the operation by outsourcing to the cloud server and verify the outsourced calculation results to ensure the accuracy of the returned calculation results. It is proved that the scheme based on the static assumption problem can achieve full secure under the standard model. Finally, through theoretical and experiment analysis, it shows that our scheme has more advantages than other schemes.

**Declaration of conflicting interests**

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

**Funding**

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work was funded by the National Natural Science Foundation of China (Grant No. 61802117), in part by the Research Foundation of Young Core Instructor in Henan province (Grant No. 2018GGJS058), and in part by the Innovative Scientists and Technicians team of Henan Provincial High Education (Grant No. 20IRTSTHN013).

**ORCID iD**

Guanghui He https://orcid.org/0000-0001-5501-6788

**References**

1. Zhao MH, Hu CY, Song XF, et al. Towards dependable and trustworthy outsourced computing: a comprehensive survey and tutorial. J Netw Comput Appl 2019; 131: 55–65.
2. Nishide T, Yoneyama K and Ohta K. Attribute-based encryption with partially hidden encryptor-specified access structures. In: Proceedings of the international conference on applied cryptography and network security (ACNS), 2009, pp.13–23. Springer, https://link.springer.com/content/pdf/10.1007%2F978-3-540-68914-0_7.pdf
3. Lai J, Deng RH and Li Y. Expressive CP-ABE with partially hidden access structures. In: Proceedings of the 7th ACM symposium on information, computer and communications security (ASIACCS), Seoul, South Korea, 2012, pp.18–19. https://ink.library.smu.edu.sg/cgi/viewcontent.cgi?article=2648&context=sis_research
4. Li J, Wang H, Zhang Y, et al. Ciphertext-policy attribute-based encryption with hidden access policy and testing. KSII Trans Internet Inform Syst 2016; 10(7): 3339–3352.
5. Yin HJ, Zhang LY and Cui YL. Improving security in ciphertext-policy attributed-based encryption with hidden access policy and testing. KSII Trans Internet Inform Syst 2019; 13(5): 2768–2780.
6. Cui H, Deng RH, Lai J, et al. An efficient and expressive ciphertext-policy attribute-based encryption scheme with partially hidden access structures, revisited. Comput Netw 2018; 133: 157–165.

7. Zhang YH, Zheng D and Deng RH. Security and privacy in smart health: efficient policy-hiding attribute-based access control. IEEE Internet Things J 2018; 5(3): 2130–2145.

8. Zhang LY, Hu GC, Mu Y, et al. Hidden ciphertext policy attribute-based encryption with fast decryption for personal health record system. IEEE Access 2019; 7: 33202–33213.

9. Xiong H, Zhao YN, Peng L, et al. Partially policy-hidden attribute-based broadcast encryption with secure delegation in edge computing. Future Gener Comput Syst 2019; 97: 453–461.

10. Shahid F, Khan A and Jeon G. Post-quantum distributed ledger for internet of things. Comput Electr Eng 2020; 83: 106581.

11. Waters B. Dual system encryption: realizing fully secure IBE and HIBE under simple assumption. In: Proceedings of the annual international cryptology conference, Santa Barbara, CA, 2009, pp.619–636, https://eprint.iacr.org/2009/385.pdf

12. Sahai A and Waters B. Fuzzy identity-based encryption. In: Proceedings of the annual international conference on the theory and applications of cryptographic techniques (EUROCRYPT), Aarhus, 2005, pp.457–473, https://eprint.iacr.org/2004/086.pdf

13. Goyal V, Pandey O, Sahai A, et al. Attribute-based encryption for fine-grained access control of encrypted data. In: Proceedings of the 13th ACM conference on computer and communications security, Alexandria, VA, 2006, pp.89–98, https://tjkopena.com/wiki/uploads/Papers/p89-goyal.pdf

14. Bethencourt J, Sahai A and Waters B. Ciphertext-policy attribute-based encryption. In: Proceedings of the IEEE symposium on security and privacy, Berkeley, CA, 2007, pp.321–334, https://www.cs.utexas.edu/~bwaters/publications/papers/cp-abe.pdf

15. Waters B. Ciphertext-policy attribute-based encryption: an expressive, efficient, and provably secure realization. In: Proceedings of the international workshop on public key cryptography, 2011, pp.53–70. Berlin: Springer, https://eprint.iacr.org/2008/290.pdf

16. Lewko A, Okamoto T, Sahai A, et al. Fully secure functional encryption: attribute-based encryption and (hierarchical) inner product encryption. In: Proceedings of the annual international conference on the theory and applications of cryptographic techniques (EUROCRYPT), 2010, pp.62–91, https://eprint.iacr.org/2010/110.pdf

17. Jin C, Feng X and Shen Q. Fully secure hidden ciphertext policy attribute-based encryption with short ciphertext size. In: Proceedings of the 6th international conference on communication and network security (ICCCNS), Singapore, 2016, pp.91–98, https://dl.acm.org/doi/10.1145/3017971.3017981

18. Zhang Y, Chen X, Li J, et al. Anonymous attribute-based encryption supporting efficient decryption test. In: Proceedings of the 8th ACM SIGSAC symposium on information, computer and communications security (ASIACCS), Hangzhou, China, 2013, pp.511–516, https://dl.acm.org/doi/10.1145/2484313.2484381

19. Zhang Y, Li J, Chen X, et al. Anonymous attribute-based proxy re-encryption for access control in cloud computing. Secur Commun Netw 2016; 9(14): 2397–2411.

20. Zhang Y, Chen X, Li J, et al. Ensuring attribute privacy protection and fast decryption for outsourced data security in mobile cloud computing. Inform Sci 2017; 379: 42–61.

21. Yan XX, Ni H, Liu Y, et al. Privacy-preserving multi-authority attribute-based encryption with dynamic policy updating in PHR. Comput Sci Inform Syst 2019; 16(3): 831–847.

22. Hao JL, Huang C, Ni JB, et al. Fine-grained data access control with attribute-hiding policy for cloud-based IoT. Comput Netw 2019; 153: 1–10.

23. Katz J, Sahai A and Waters B. Predicate encryption supporting disjunctions, polynomial equations, and inner products. In: Proceedings of the annual international conference on the theory and applications of cryptographic techniques (EUROCRYPT), 2008, pp.146–162, https://www.cs.umd.edu/~jkatz/papers/ec08.pdf

24. Green M, Hohenberger S and Waters B. Outsourcing the decryption of ABE ciphertexts. In: Proceedings of the 20th USENIX conference on security, San Francisco, CA, 2011, pp.1–16, https://static.usenix.org/events/sec11/tech/full_papers/Green.pdf

25. Li L, Zhang YH, Chen XF, et al. Secure attribute-based data sharing for resource-limited users in cloud computing. Comput Secur 2018; 72: 1–12.

26. Fan K, Wang JX, Wang X, et al. A secure and verifiable outsourced access control scheme in fog-cloud computing. Sensors 2017; 17(7): 1695–1710.

27. Zhang R, Ma H and Lu Y. Fine-grained access control system based on fully outsourced attribute-based encryption. J Syst Softw 2017; 125(3): 344–353.

28. Zhao ZY, Wang JH, Xu KY, et al. Fully outsourced attribute-based encryption with verifiability for cloud storage. J Comput Res Develop 2019; 56(2): 442–452.

29. Yan XX, He X, Yu JX, et al. White-box traceable ciphertext-policy attribute-based encryption in multi-domain environment. IEEE Access 2019; 7: 128298–128312.