Hindawi
Journal of Sensors
Volume 2020, Article ID 8861114, 11 pages
https://doi.org/10.1155/2020/8861114

Research Article

Offline/Online Outsourced Attribute-Based Encryption with Partial Policy Hidden for the Internet of Things

Xixi Yan, Guanghui He, Jinxia Yu, Yongli Tang, and Mingjie Zhao

School of Computer Science and Technology, Henan Polytechnic University, Jiaozuo, 454000 Henan, China

Correspondence should be addressed to Yongli Tang; yltang@hpu.edu.cn

Received 19 April 2020; Revised 9 July 2020; Accepted 31 July 2020; Published 4 September 2020

Academic Editor: Fei Yu

In the Internet of Things (IoT) environment, the intelligent devices collect and share large-scale sensitive personal data for a wide range of application. However, the power of storage and computing of IoT devices is limited, so the mass perceived data will be encrypted and transmitted to a cloud platform-interconnected IoT devices. Therefore, the concern how to save the encryption/decryption cost and preserve the privacy of the sensitive data in IoT environment is an issue that deserves research. To mitigate these issues, an offline/online attribute-based encryption scheme that supports partial policy hidden and outsourcing decryption will be proposed. This scheme adopts offline/online attribute-based encryption algorithms; then, the key generation algorithm and encryption algorithm are divided into two stages: offline stage and online stage. Meanwhile, in order to solve the problem of policy disclosure under the cloud platform, the policy hidden is supported, that is, the attribute is divided into the attribute value and the attribute name. For the pairing operation involved in decryption process, a verifiable outsourced decryption is implemented. Our scheme is constructed based on composite bilinear groups, which meets full security under the standard model. Finally, by comparing with other schemes in terms of functionality and computational overhead, it is shown that the proposed scheme is more efficient and applicable to the mobile devices with limited computing and storage functions in the Internet of Things environment.

1. Introduction

With the continuous development of the Internet of Things technology, it has been widely used in the fields of health care, smart home, industrial manufacturing, and environmental monitoring. But the computing and storage resources of Internet of Things equipment are often limited; an increasing number of individuals or organizations are outsourcing the storage of personal information to the cloud server to achieve lower cost. However, due to the cloud server being not completely trusted, therefore, how to protect the private information contained in the data and how to deal with the huge computing cost for the mobile devices with limited resources are the problems that should be solved in the current research.

In the application of intelligent medicine, personal health information records are collected through wearable devices (e.g., smart bracelets); then, it will be solved by a medical information integration platform. Personal Heath Record (PHR) is the core basic component of intelligent medical, which involves a lot of personal privacy information of users. The data need to be shared with relevant doctors, relatives, and friends, so it is important to achieve the fine-grained access control of data and related equipment. Sahai and Waters proposed a new public key cryptosystem called attribute-based encryption (ABE) [1]. Subsequently, it can be divided into two categories according to the location of the access policy: key policy attribute-based encryption (KP-ABE) [2] and ciphertext policy attribute-based encryption (CP-ABE) [3]. In CP-ABE schemes, access policy is embedded in ciphertext implicitly and outsourced to Cloud Service Provider (CSP) together with ciphertext in cloud environment. Because access policies are publicly available, everyone can access policies that contain some private information. For example, in the intelligent medical system, the patient authorized the cardiologist to access the encrypted data through the access policy as {Department: Cardiology; Doctor: Alice}. If anyone sees the encrypted data, it would

still be concluded that the patient obtained "heart disease" without decryption. If the content of the attribute value in the policy is not visible, that is, the policy is set as {Department: ×××; Doctor: ×××}, then the patient's privacy can be guaranteed.

In order to avoid leaking the sensitive information implicit in the strategy, how to hide the access strategy has become a concern of many scholars. Nishide et al. [4] first proposed the ciphertext policy attribute-based encryption with hiding access structure. The implicit access structure in the ciphertext is not sent along with the ciphertext, that is, no one can obtain the access structure information. But the policy in the scheme only supports the "AND" gate structure. Lai et al. [5] proposed a CP-ABE with partial access structure hidden which achieved better policy expression. Different from the previous schemes, the attribute is divided into two parts: attribute name and attribute value. The attribute name can be publicized, while the attribute value is hidden. Li et al. [6] proposed an efficient attribute-based encryption scheme with partial hiding 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. [7] proposed a more efficient scheme for the deficiency of Li et al.'s [6] scheme in the standard model. It is successfully reduced to the DBDH assumption. Cui et al. [8] constructed a scheme based on a composite order group supporting hidden attribute value, but it only achieves selective security. In the application scenario of the electronic medical system, Zhang et al. [9] not only implements the policy semiconcealment but also takes less computing cost and storage overhead during the decryption process. In addition, the scheme is fully secure under the standard model. However, the bilinear pairing operation and modular power operation involved in decryption process are still associated with numbers of attribute. For decreasing the complicate pairing operation, Hu et al. [10] proposed a semihiding attribute-based encryption scheme with constant pairing operation, but the modular power operation is still linearly related to the number of attributes. However, the above scheme only realizes access structure hidden, and the computing overhead relates to the complexity of access structure and the number of attributes; what is more, the process of encryption and decryption also needs a large number of modular power operation and pairing operation.

It is a fact that IoT devices need to be in real-time online when generating policy-related ciphertext, but the IoT devices with limited computing and storage are not always online. In order to solve this problem, Li et al. [11] put forward an offline/online attribute-based encryption supporting the access policy invisible. The key generation and encryption operation are divided into offline and online phases. That is to say, the key and ciphertext are precalculated in the offline phase, while a small amount of overhead is calculated to complete all the key components and ciphertext in the online phase. However, the pairing operation involved in decryption is still linearly related to the number of attributes required for decryption. In this paper, we propose an offline/online attribute-based encryption scheme which can not only hide access structure but also support outsourcing decryption. The main contributions are as follows.

(1) Partial access policy hidden: different from the technology of hiding access structure adopted by Li et al. [11], it divides attributes into two parts: attribute name and attribute value. Attribute name can be disclosed, and attribute value can be hidden. Hence, attribute name can be uploaded to cloud server provider (CSP) together with ciphertext, but attribute value is not visible

(2) Outsourced decryption: in the decryption process, the bilinear pair operation and modular power operation are outsourced to the CSP for execution. The user only needs to verify the returned results and perform constant exponential operation to recover the plaintext

(3) Fully secure: in this paper, our scheme is based on composite order groups and proved to be fully secure by the dual-system encryption technology [12]

(4) Performance advantages: by comparing with the previous schemes from the aspects of function and performance, the proposed scheme has more advantages. And it is shown that our scheme is feasible in IoT by carrying out simulation experiments based on the PBC function library.

2. Related Work

2.1. Policy Hidden. In order to preserve the privacy of user attributes in the cloud environment, Nishide et al. [4] first proposed a CP-ABE scheme with access structure hidden. Lewko et al. [13] proposed a fully secure CP-ABE scheme by using a dual-system encryption technology under the standard model. Subsequently, Lewko and Waters [14] put forward a new proof method to achieve full security; however, the efficiency is lower. Lai et al. [5] and Jin et al. [15] gave a CP-ABE scheme supporting partial policy hidden. The scheme is proved to satisfy fully secure, but the access structure only support the "AND" gate structure. In order to reduce the bilinear pairwise operation and modular exponentiation involved in decryption process, the schemes [5, 16] gave the specific scheme. 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 the scheme [16] only supports the "AND" gate structure, and the linear pair operation and modular exponentiation are still linearly related to the number of attributes during decryption period. At the same time, the scheme is proved to be selective secure, while Lai et al. [5] adopts more flexible LSSS structure, and the scheme is fully secure under the standard model. But the bilinear pairing operations and modular exponentiations involved in the user testing phase and the decryption phase increase linearly with the complexity of the policy. Yan et al. [17] introduced a multiauthority attribute-based encryption of partial policy hidden with dynamic policy updating. In the scheme, the policy

hiding is only to hide the attribute value, so it is called semi-hidden policy. The function of hidden attribute completely can also be realized by the inner product predicate encryption technology [18], but most of them only support the "AND" gate structure with weak expression ability, so there is a range of limitation in the actual application process.

2.2. Offline/Online Attribute-Based Encryption. The offline/online encryption, namely, preprocesses a lot of heavy work in the offline phase and responds to key requests or encryption tasks rapidly in the online phase. Even et al. [19] first proposed the offline/online digital signature technology. Liu et al. [20] gave an identity-based offline/online signature scheme in a wireless sensor network environment. Guo et al. [21] proposed an identity-based offline/online encryption scheme. Most of the computational work is preprocessed in the offline stage, and the actual encryption operation is completed in the online stage. Hohenberger and Waters [22] introduced an offline/online attribute-based encryption scheme in 2014, which is the first scheme that adopted the offline/online technology. Liu et al. [23] proposed a new ciphertext attribute-based encryption scheme by combining the offline/online technology and verification outsourcing technology. Wang et al. [24] proposed an offline/online attribute-based encryption scheme that achieved full security under the standard model. However, there was no verification of the part decryption results which is completed by cloud. Among existing intrusion prevention systems available, an industrial network intrusion detection algorithm is proposed based on the multifeatured data clustering optimization model [25]. With the development of electronic chip technologies of IoT, Liang et al. [26] introduced a fast deep reinforcement learning- (DRL-) based detection algorithm for virtual IP watermarks, by combining the technologies of mapping function and DRL to preprocess the ownership information of the IP circuit resource.

2.3. Outsourced Attribute-Based Encryption. Green et al. [27] proposed the first outsourced attribute-based encryption scheme which is secure in a random oracle model. The scheme commits the decryption operation to the decrypt server provider, so the ciphertext was converted into the type by ElGamal encryption, and then delivered to users so as to reduce computing cost of data user. Lai et al. [28] realized the outsourcing decryption and provided the accuracy verification of outsourcing calculation. Li et al. [29] presented an offline/online attribute-based encryption scheme, which will reduce the computational overhead during encryption phase with the offline/online technology. Also, the "chameleon" hash function was introduced to implement verification before the decryption phase. What is more, the scheme was proved to satisfy the adaptive chosen ciphertext attack security, but the bilinear pairing operation involved in decryption procession was still a large overhead for the user. Fan et al. [30] introduced a verifiable outsource scheme for multi-authorization in cloud-fog computing, which outsources encryption and decryption to fog nodes closed to the end user. Relative to the remote cloud sever provider, fog nodes can handle data with low latency, which was an ideal choice for real-time calculation of data. Zhang et al. [31] 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 verification mechanism. Zhao et al. [32] put forward a verifiable full outsourcing scheme based on the original scheme [31]. The scheme supports verifiable and optimized performance that the computational cost does not increase significantly with the number of attributes or access policy complexity. Yu et al. [33] introduced a verifiable outsourced attribute-based encryption with partial policy hidden. In the particularity of blockchain-based industrial network, the data storage management faces enormous challenges. Liang et al. [34] focuses on data security issues in the industrial network and designs a storage and repair scheme for fault-tolerant data coding.

3. Preliminaries

3.1. Composite Order Bilinear Group. The proposed scheme is based on the composite order bilinear group whose order is the product of three distinct primes. Let Φ be an algorithm that inputs security parameter 1^λ and outputs a tuple $(p_1, p_2, p_3, G, G_T, e)$, where p_1, p_2, p_3 are 3 distinct primes, G, G_T are cycle groups of order $N = p_1p_2p_3$, and $e : G \times G \longrightarrow G_T$ is a map function such that

(1) Bilinear:

$$\forall g, h \in G, a, b \in \mathbb{Z}_N, e\left(g^a, h^b\right) = e(g, h)^{ab} \quad (1)$$

(2) Nondegenerate: 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, G_T and the mapping function e, it is computable in polynomial time in λ. Let $G_{p_1}, G_{p_2}, G_{p_3}$ represent subgroups of G, the subgroups have order p_1, p_2, p_3, respectively, then $G = G_{p_1} \times G_{p_2} \times G_{p_3}$. 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.

3.2. Linear Secret Sharing Scheme (LSSS). The secret sharing scheme on the participant set P is called the linear secret sharing scheme if the following conditions are met.

(1) A vector can be formed by the secret share of each party over $\mathbb{Z}_p$

(2) For the secret sharing scheme Π, there is a matrix $\boldsymbol{M}$ of size $\ell \times n$ that maps each row of the matrix to an associated participant P. For $i = 1, \cdots, \ell$, $\rho(i)$ is the party associated with the i-th row of $\boldsymbol{M}$. We first generate a column vector $\boldsymbol{v} = (s, y_2, y_3, \cdots, y_n)$, where $s \in \mathbb{Z}_p$ is a shared secret andr_i is randomly selected, $i = 2, \cdots, n$. According to scheme Π, $\mathbf{Mv}$ is ℓ secret shares of the shared secret s, which indicates $\lambda_i =$

$(\boldsymbol{Mv})_i$ is held the secret share by the participants $\rho(i)$.

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

3.3. Key Derivation Function (KDF). KDF algorithm outputs bit string by inputting original secret key DK and length l. KDF algorithm is secure if it has following negligible advantage for adversary in any probability polynomial time.

$$|\Pr[\mathscr{A}(KDF(DK, l)) = 1] - \Pr[\mathscr{A}(\theta) = 1]|. \quad (2)$$

Note that $\theta \in \{0, 1\}^l$.

3.4. Complexity Assumption. The order of group G is defined as the product of three different prime numbers. For any nonempty set $Z \subseteq \{1, 2, 3\}$, the order of subgroup of group G is $\prod_{i\in Z} p_i$. In this paper, the subgroup is denoted by G_Z. The security is based on the following complexity assumptions, and a detailed description of the complexity assumptions is given.

$$G : (N = p_1p_2p_3, G, G_T, e) \xleftarrow{R} \Phi, g_{Z_2} \xleftarrow{R} G_{Z_2}, \cdots, g_{Z_k} \xleftarrow{R} G_{Z_k},$$
$$D : \left(G, g_{Z_2}, \cdots, g_{Z_k}\right), T_0 \xleftarrow{R} G_{Z_0}, T_1 \xleftarrow{R} G_{Z_1}. \quad (3)$$

Assumption 1. (The General Subgroup Decision Assumption): Given a group generation algorithm Φ, $Z_0, Z_1, Z_2, \cdots, Z_k$ represent a nonempty subset of a set $\{1, 2, 3\}$, where Z_i satisfies $Z_0 \cap Z_i \neq \varnothing \neq Z_1 \cap Z_i$ or $Z_0 \cap Z_i = \varnothing = Z_1 \cap Z_i$, $i \geq 2$. Define the following distribution:

If the advantage of Adversary $\mathscr{A}$ satisfies $Adv^1_{\Phi,\mathscr{A}}(\lambda) = |\Pr[\mathscr{A}(D, T_0) = 1] - \Pr[\mathscr{A}(D, T_1) = 1]|$, then this assumption can be broken.

$$G : (N = p_1p_2p_3, G, G_T, e) \xleftarrow{R} \Phi, g_1 \xleftarrow{R} G_{p_1}, g_2, X_2,$$
$$Y_2 \xleftarrow{R} G_{p_2}, g_3 \xleftarrow{R} G_{p_3}, \alpha, s \xleftarrow{R} \mathbb{Z}_N,$$
$$D : (G, g_1, g_2, g_3, g_1^{\alpha}X_2, g_1^{s}Y_2), T_0 = e(g, g)^{\alpha s}, T_1 \xleftarrow{R} G_T. \quad (4)$$

Definition 2. For any probability polynomial time, if $Adv\,1_{\Phi,\mathscr{A}}(\lambda)$ is a negligible function, then the algorithm meets Assumption 1.

Assumption 3. Given a group generation algorithm Φ, define the following distribution:

If the advantage of Adversary $\mathscr{A}$ satisfies $Adv^2_{\Phi,\mathscr{A}}(\lambda) = |\Pr[\mathscr{A}(D, T_0) = 1] - \Pr[\mathscr{A}(D, T_1) = 1]|$, then this assumption can be broken.

Definition 4. For any probability polynomial time, if $Adv^2_{\Phi,\mathscr{A}}(\lambda)$ is a negligible function, then the algorithm meets Assumption 3.

4. System Algorithm and Security Model

4.1. Algorithm Definition. The algorithm included in this scheme is composed of the following seven algorithms:

$Setup(\lambda, U) \longrightarrow PK, MSK$: the algorithm inputs the security parameters λ, attributes universe U, and outputs master key MSK and the public parameter PK including the Key Derivation Function (KDF)

$Offline.KeyGen(PK, \varsigma) \longrightarrow IK$: this algorithm is implemented by the attribute authority in the offline phase. Input the public parameter PK and attribute set ς, and return the intermediate key IK

$Online.KeyGen(PK, MSK, IK, \varsigma) \longrightarrow SK, TK$: this algorithm is implemented by the attribute authority in the online phase. It inputs the public parameter PK, master key MSK, attribute set ς, and intermediate key IK, then returns the transformed key $TK$and secret key SK, where TK is used for outsourced decryption and SK is used for user local decryption

$Offline.Enc(PK, A) \longrightarrow IC$: the algorithm is run by the data owner in the offline stage. Input public parameters PK and access policy A; it will output intermediate ciphertext IC

$Online.Enc(PK, (M, \rho), IC, m) \longrightarrow CT$: the algorithm is run by the data owner in the online stage. Input public parameters PK, intermediate ciphertext IC, and message m; then, it outputs complete ciphertext CT

$Transform_{\text{out}}(TK, CT) \longrightarrow CT'$: the algorithm is executed by the cloud server provider (CSP) to generate partial decryption ciphertext CT' by inputting the transformed key TK and the ciphertext CT

$Decrypt(SK, CT, CT', PK) \longrightarrow m$: the algorithm is executed by the local user to generate m by inputting secret key SK, complete ciphertext CT, and partial decryption ciphertext CT', then returns m.

4.2. Security Model. We define the security model of this paper through the security game between Challenger (Simulator) $\mathscr{B}$ and Adversary $\mathscr{A}$. The game process is as follows:

Setup. Challenger $\mathscr{B}$ performs the Setup algorithm and outputs the public parameter PK to the Adversary $\mathscr{A}$

Phase 1. Challenger $\mathscr{B}$ initializes empty table T, empty set D, and integer $i = 0$. Adversary $\mathscr{A}$ can repeatedly ask any of the following queries:

Create (ς). The challenger sets $i = i + 1$ run the key generation algorithm on the attribute set S to obtain the key set (SK, TK), and finally stores(i, ς, SK, TK) in table T

Corrupt (x). If there is an x-th entity in table T, then the challenger obtains the entity (x, ς, SK, TK) and sets $D \coloneqq D \cup \{\varsigma\}$, and then outputs the key set (SK, TK) to Adversary $\mathscr{A}$. If it does not exist, then outputs "$\perp$"

Challenge. For all $\varsigma \in D, \varsigma \notin A^*$, Adversary $\mathscr{A}$ submits two equal-length messages m_0^*, m_1^* and access structures A^* to B, the Challenger $\mathscr{B}$ selects $b \in \{0,1\}$and encrypts the messages m_b in the access structure A^*, then sends the generated ciphertext $CT^*$to the Adversary $\mathscr{A}$.

Phase 2. The Challenger $\mathscr{B}$ continues to respond to the adversary's queries in the way of *Phase 1*, but the adversary cannot ask the challenger the attribute set ςthat satisfies the policyA^*

Guess. The Adversary $\mathscr{A}$ outputs the guess value $b' \in \{0,1\}$, and if $b' = b$, then the Adversary $\mathscr{A}$ wins the game.

5. Our Construction

The offline/online attribute-based encryption scheme which supports the partial policy hidden and outsourced decryption proposed in this paper is inspired based on references [14, 24] and consists of the following seven algorithms. The scheme is constructed as follows:

$Setup(\lambda, U) \longrightarrow PK, MSK$: the algorithm inputs the security parameters λ, attributes universe U, and selects a linear group G of order $N = p_1p_2p_3$, where $U = \mathbb{Z}_N, p_1, p_2, p_3$ are three different prime numbers, and p_i represents the order of subgroup G_{p_i}. Then, it randomly selects$\alpha, a, k, u, r \in \mathbb{Z}_N$ and $g \in G_{p_1}$, meanwhile setting the key derivation function KDF with the output length l and the resistant-collusion hash function $H : \{0,1\}^* \longrightarrow \mathbb{Z}_N$, and finally outputs the public parameters $PK = (N, g, g^a, g^k, e(g,g)^\alpha, u, r, KDF, l, H)$ and the master key $MSK = (g^\alpha, g_3 \in G_{p_3})$.

The user attribute set is defined as $\varsigma = (\chi_S, S)$, where χ_S represents the attribute name index, $\chi_S \subseteq \mathbb{Z}_N$, and $S = \{s_i\}_{i\in\chi_S}$ represents the attribute value set

$Offline.KeyGen(PK, \varsigma) \longrightarrow IK$: the algorithm selects $t' \in \mathbb{Z}_N$ and calculates$K_i' = (g^{s_i})^{t'}$, where$i \in \chi_S$, then outputs $IK = (\{K_i'\}_{i\in\chi_S}, t')$

$Online.KeyGen(PK, MSK, IK, \varsigma) \longrightarrow SK, TK$: it randomly selects $h, z \in \mathbb{Z}_N$,$R, R', R'', \{R_i\}_{i\in\chi_S} \in G_{p_3}$ and calculates $\tilde{K} = g^\alpha g^{at} g^{hk} R, \tilde{K}' = g^h R', \tilde{K}'' = g^{t'} R'', \tilde{K}_i = K_i'$, then outputs the transformed key $TK = (K = \tilde{K}^{1/z} = g^{(\alpha+at+hk)/z} R^{1/z}, K' = \tilde{K}'^{1/z} = g^{h/z} R'^{1/z}, K'' = \tilde{K}''^{1/z} = g^{t'/z} R''^{1/z}, K_i = \tilde{K}_i = g^{s_i t'/z})$ and user secret key $SK = (z, TK)$.

$Offline.Enc(PK, A) \longrightarrow IC$: the algorithm inputs the specified access policy$A = (\mathbf{M}, \rho, \Psi)$, where M is a matrix of $\ell \times n$ and $\rho$is a function that maps the x-th row of the matrix $\mathbf{M}_x$ to the attribute name index. And $\Psi = (t_{\rho(1)}, t_{\rho(2)}, \cdots t_{\rho(\ell)}) \in \mathbb{Z}_N^\ell$ is the attribute value set associated with access policy $(\mathbf{M}, \rho)$. Then, it selects a vector $v = (s, v_1, v_2, \cdots v_n)$, $r_x \in_R \mathbb{Z}_N$, computes$key = e(g,g)^{\alpha s}, C = g^s, C'' = (g^k)^s$, $\{C_{1,x} = g^{a\cdot M_x \cdot v} (g^{t_{\rho(x)}})^{-r_x}, D_{1,x} = g^{r_x}\}_{x\in[1,\ell]}$, and finally generates intermediate ciphertext $IC = (key, (M, \rho), C, C''\{C_{1,x}, D_{1,x}\}_{x\in[1,\ell]})$

$Online.Enc(PK, IC, m) \longrightarrow CT$: it firstly computes $C' = m \cdot key$. Next, it selects$t \in G_T$ and computes $\bar{C} = t \oplus KDF(key, l)$, $\hat{C} = u^{H(m)} r^{H(t)}$, then finally outputs the complete ciphertext $CT = ((M, \rho), C, C', C'', \{C_{1,x}, D_{1,x}\}_{i\in[1,\ell]}, \bar{C}, \hat{C})$

$Transform_{\text{out}}(TK, CT) \longrightarrow CT'$: after receiving the transformed key TK and the ciphertext CT, if the attribute set ς satisfies access policy A, there is a subset $\chi \in I_{(M,\rho)}$ that satisfies $\{\rho(i) \mid i \in \chi\} \subseteq \chi_S$, where $I_{(M,\rho)} \subseteq \{1, 2, \cdots, \ell\}$ denotes the subset of $\{1, 2, \cdots, \ell\}$ that meets $(\mathbf{M}, \rho)$, and then there exists a set of constants $\{\omega_i\}_{i\in\chi}$ such that $\sum_{i\in\chi} \omega_i \lambda_i = s$ holds, and λ_i is the valid share of the secret s. The procedure of transformed ciphertext $CT_{transform}$ follows the steps below:

$$CT_{\text{transform}} = \frac{e(C,K)e\left(C', K'\right)}{\Pi_{i\in\chi}\left(e\left(C_{1,i}, K''\right)e\left(D_{1,i}, K_{\rho(i)}\right)\right)^{\omega_i}} = \mathrm{e}(\mathrm{g}, \mathrm{g})^{\alpha s/z}. \tag{5}$$

which finally gain the partial decryption ciphertext $CT' = (CT_{\text{transform}}, C', \bar{C}' = \bar{C}, \hat{C}' = \hat{C})$

$Decrypt(SK, CT, CT', PK) \longrightarrow m$: the algorithm is run by data user, which takes $CT' = (CT_{\text{transform}}, C', \bar{C}' = \bar{C}, \hat{C}' = \hat{C})$ and SK as input, then computes $key = C/CT_{\text{transform}}^z = e(g,g)^{\alpha s}$, $m \longleftarrow C/key$. If$\bar{C} \oplus KDF(key, l) \longrightarrow t$ and $\hat{C}' = u^{H(m)} r^{H(t)}$ hold, meanwhile if $b = 1$, it outputs m, else $b = 0$, then returns "$\perp$."

6. Security Proof

$$\begin{aligned} K &= \left((g_1^\alpha X_2) g_1^{at+hk}\right)^{1/z} R^{1/z} g_2^{\gamma/z}, K' = g_1^{h/z} R'^{1/z} g_2^{\gamma/z}, K'' \\ &= g_1^{t/z} R''^{1/z}, K_i = g^{s_i t'/z} R_i^{1/z}. \end{aligned} \tag{6}$$

Theorem 5. *If the Assumption 1 and Assumption 3 hold, then the proposed scheme based on the defined security model is fully secure and satisfies CPA (Chosen-Plaintext Attack) security.*

Proof. The security proof of the scheme is similar to that in literature [14], that is, the dual-system encryption technology is used to prove its security. First, define two semifunctional structures: semifunctional ciphertext and semifunctional key. The normal secret key can decrypt normal ciphertext and semifunctional ciphertext, but the semifunctional secret key cannot decrypt semifunctional ciphertext. And semifunctional key and semifunctional ciphertext are only used in security proof and do not appear in actual systems.

Semifunctional key: it first calls the normal key generation algorithm to generate a normal key $K, K', K'', \{K_i\}_{i\in\chi_S}$ and then randomly selects elements$\eta, \eta' \in G_{p_2}$ to generate a semifunctional key: $K\eta, K'\eta', K'', \{K_i\}_{i\in\chi_S}$; in other words, except for K, K', the remaining components are multiplied by the elements inG_{p_2}.

Semifunctional ciphertext: first call the normal encryption algorithm to generate a normal ciphertext: C, C', C'', $\{C_{1,x}, D_{1,x}\}_{x\in[1,\ell]}$, then select a random exponent $a', k', s' \in \mathbb{Z}_N$, and random vector $\omega \in \mathbb{Z}_N$, where s' is the first element

in the set, random exponent $\eta_i, \gamma_x \in \mathbb{Z}_N$, then the semifunctional ciphertext is $C, C' g_2^{s'}, C'' g_2^{s' k'}, \{C_{1,x} g_2^{a' M_x \omega} g_2^{-\eta_{\rho(x)} \gamma_x}, D_{1,x} g_2^{\gamma_x}\}_{x \in [1,\ell]}$. The element structure in group G_{p_2} here is similar to the element structure in G_{p_1}, but not related to public parameters.

First, let Q denote the total number of key queries made by the adversary, and define the game $Game_k$, where$k \in [0, Q]$.

$Game_k$: in this game, the ciphertext obtained by to the attacker is a semifunctional ciphertext, the first k keys are also semifunctional, and the remaining keys are normal.

The security proof of the scheme based on Assumption 1 and Assumption 3 is demonstrated through a series of games. We first transition from $Game_{real}$ to $Game_0$, then to $Game_1$, and until to $Game_Q$, where the key and ciphertext submitted to the attacker are semifunctional. Finally, to $Game_{final}$ stop, the ciphertext obtained by to the attacker at this time is generated by semifunctional encryption of random messages. Because the attacker does not have any advantages in the final game, the security proof of the scheme in this paper ends here.

Lemma 6. *Under Assumption 1 (the general subgroup decision assumption), no polynomial time attacker can achieve a nonnegligible difference in advantage between $Game_{real}$ and $Game_0$.*

Proof. We first create an algorithm $\mathcal{B}$ in probabilistic polynomial time to break the general subgroup decision assumption and set $Z_0 := \{1\}, Z_1 := \{1, 2\}, Z_2 := \{1\}, Z_3 := \{3\}$. Let g_1, g_3, T input to Algorithm $\mathcal{B}$, where g_1 is the generator of the group G_{p_1}, $g_3$is the generator of the group G_{p_3}, and T is the random element of the group G_{p_1} or the random element of the group $G_{p_1 p_2}$. $\mathcal{B}$ can act as a simulator to interact with the adversary, and $\mathcal{B}$ can simulate or interact with the Adversary $\mathcal{A}$, depending on the nature of T.

$\mathcal{B}$ chooses a random exponent $\alpha, a, k \in \mathbb{Z}_N$ and sets public parameters $PK = (N, g = g_1, g^a = g_1^a, g^k = g_1^k, e(g, g)^\alpha = e(g_1, g_1)^\alpha, u, r, KDF, l, H)$ to submit PK to Adversary $\mathcal{A}$. We notice that Simulator $\mathcal{B}$ knows the master key MSK. When $\mathcal{A}$ makes a secret key query, $\mathcal{B}$ will call the normal key generation algorithm to create a secret key.

The adversary requests a challenge ciphertext and message related to the access policy $A = (\mathbf{M}, \rho, \Psi)$. $\mathcal{B}$ randomly selects bit b and generates the ciphertext m_b; then, $g^s$of implicit setting is equivalent to the part ofG_{p_1} in T. $\mathcal{B}$ randomly selects the vector $\tilde{v} \in Z_N^n$, and the first element of the vector has a value of 1. At the same time, let $v = s\tilde{v}$ and select $r_x \in \mathbb{Z}_N, x \in [1, \ell]$ randomly, then set $r_x = s\tilde{r}_x$. We should note that the elementss, v, r_x are distributed randomly, and then, the corresponding ciphertexts are $C = T, C' = m_b \cdot e(g_1, T)^\alpha$, $C'' = T^k, C_{1,x} = T^{a \cdot M_x \cdot \tilde{v}} T^{-\tilde{r}_x \cdot \eta_{\rho(x)}}, D_{1,x} = T^{\tilde{r}_x}$.

If $T \in G_{p_1}$, the ciphertext is normal ciphertext, and $\mathcal{B}$ simulates the game $Game_{real}$ and interacts with $\mathcal{A}$. If $T \in G_{p_1 p_2}$, it is a semifunctional ciphertext. And the elements in G_{p_2} are set as follows: $g^{s'}$ is the components of G_{p_2} in T, k' is equivalent to the value of $k \mod p_2$, a' is equivalent to $a \mod p_2$, ω is equivalent to $s\tilde{v} \mod p_2$, $\eta_{\rho(x)}$ is equivalent to $t_{\rho(x)} \mod p_2$, and γ_x is equivalent to $s'\tilde{r} \mod p_2$. We note that these values are generated by proper distribution, and the values of the element $\mod p_1, p_2$ uniformly selected at random $\mod N$ are independently and uniformly distributed. We also notice that public parameters will leak the value of $a, k \mod p_1$, so when $T \in G_{p_1 p_2}$, $\mathcal{B}$ and $\mathcal{A}$ simulate game$Game_0$, and then $\mathcal{B}$ can use adversary's nonnegligible distinction in these games to obtain a nonnegligible advantage to break Assumption 1 (general subgroup decision assumption).

Lemma 7. *Under Assumption 1 (the general subgroup decision assumption), no polynomial time attacker can achieve a nonnegligible difference in advantage between $Game_{k-1}$ and $Game_k$.*

Proof. We first create an algorithm $\mathcal{B}$ in probabilistic polynomial time to break the general subgroup decision assumption and set $Z_0 := \{1, 3\}, Z_1 := \{1, 2, 3\}, Z_2 := \{1\}g$, $Z_3 := \{3\}, Z_4 := \{1, 2\}, Z_5 := \{2, 3\}$. Let $g_1, g_3, X_1 X_2, Y_2 Y_3$, T input to Algorithm $\mathcal{B}$, where g_1, X_1 is the generator of the group G_{p_1}, X_2, Y_2 is the generator of the group G_{p_2}, $g_3, Y_3$is the generator of the group G_{p_3}, and $T$is the random element of the group G_{p_1} or the random element of the group $G_{p_1 p_2 p_3}$. And $\mathcal{B}$ can simulate $Game_{k-1}$ or $Game_k$ to interact with Adversary $\mathcal{A}$, depending on the nature of T.

$\mathcal{B}$ chooses a random exponent $\alpha, a, k \in \mathbb{Z}_N$ and sets public parameters $PK = (N, g = g_1, g^a = g_1^a, g^k = g_1^k, e(g, g)^\alpha = e(g_1, g_1)^\alpha, u, r, KDF, l, H)$ to submit PK to Adversary $\mathcal{A}$. We noticed that Simulator $\mathcal{B}$ knows the master key MSK. When $\mathcal{A}$ makes a secret key request, $\mathcal{B}$ will call the normal key generation algorithm to create a private key in response to $\mathcal{A}$'s key query. In response to the first $k-1$ key query of a, $\mathcal{B}$ generates a semifunctional key according to the following. First, the normal key generation algorithm is called to generate the normal key$K, K', K'', \{K_i\}_{i \in \chi_S}$, and then the random value τ, τ' is selected to generate the semifunctional key: $K (Y_2 Y_3)^\tau, K' (Y_2 Y_3)^{\tau'}, K'', \{K_i\}_{i \in \chi_S}$, where group elements $Y_2^\tau, Y_2^{\tau'}$ are distributed uniformly and randomly in G_{p_2}.

In order to generate semifunctional challenge ciphertext, the adversary requests a challenge ciphertext and message m_0, m_1 related to access policy$A = (\mathbf{M}, \rho, \Psi)$. $\mathcal{B}$ randomly selects bit b and generates the ciphertext of m_b, and the g^s is equivalent to the part of G_{p_1}in T. $\mathcal{B}$ randomly selects the vector $\tilde{v} \in Z_N^n$ and the first element value of the vector is 1, and set $g^s = X_1, v = s\tilde{v}, g^{r_x} = X_1^{\tilde{r}_x}$; then, the corresponding ciphertext calculated is as follows: $C = X_1 X_2, C' = m_b \cdot e(g_1, X_1 X_2)^\alpha, C'' = (X_1 X_2)^k, C_{1,x} = (X_1 X_2)^{a \cdot M_x \cdot \tilde{v}} (X_1 X_2)^{-\tilde{r}_x \cdot \eta_{\rho(x)}}$, $D_{1,x} = (X_1 X_2)^{\tilde{r}_x}$, where set $g_2^{s'} = X_2, k' = k \mod p_2, a' = a \mod p_2, \eta_{\rho(x)} = t_{\rho(x)} \mod p_2, g_2^{r_x} = X_2^{\tilde{r}_x}$ implicitly. In order to create a semifunctional ciphertext, the value of $a, k \mod$

p_2 will not be revealed by the public parameters. To generate the k-th key request query for the associated attribute set, $\mathcal{B}$ randomly selects $t, z \in \mathbb{Z}_N$ and the random element $R, R', R'', \{R_i\} \in G_{p_3}$ and calculates the following components: $K = (g_1^{\alpha+at}T^k)^{1/z}R^{1/z}, K' = T^{1/z}R'^{1/z}, K'' = g^{t/z}R''^{1/z}, K_i = g^{s_i t'/z}R_i^{1/z}$.

If $T \in G_{p_1p_3}$, the distributed key is a normal key. If $T \in G_{p_1p_2p_3}$, the distributed key is a semifunctional key, so when $T \in G_{p_1p_3}$, $\mathcal{B}$ simulates the game $Game_{k-1}$ to interact with adversary. When $T \in G_{p_1p_2p_3}$, $\mathcal{B}$ simulates the game $Game_k$. Then, $\mathcal{B}$ can take advantage of adversary's nonnegligible difference in these games to obtain a nonnegligible advantage to break Assumption 1 (general subgroup decision assumption).

Lemma 8. *Under Assumption 3 (the general subgroup decision assumption), no polynomial time attacker can achieve a nonnegligible difference in advantage between $Game_Q$ and $Game_{final}$.*

Proof. We first create an algorithm $\mathcal{B}$ in probabilistic polynomial time to break the general subgroup decision Assumption 3. Let$g_1, g_2, g_3, g_1^\alpha X_2, g_1^s Y_2, T$ input to Algorithm $\mathcal{B}$, where T is the random element of$e(g_1, g_2)^{\alpha s}$ or the group G_T. And $\mathcal{B}$ can simulate $Game_{k-1}$ or $Game_k$ to interact with the Adversary $\mathcal{A}$, depending on the nature of T.

$\mathcal{B}$ chooses a random exponent $\alpha, a, k \in \mathbb{Z}_N$ and sets public parameters $PK = (N, g = g_1, g^a = g_1^a, g^k = g_1^k, e(g, g)^\alpha = e(g_1, g_1)^\alpha, u, r, KDF, l, H)$ to submit PK to Adversary $\mathcal{A}$. We noticed that Simulator $\mathcal{B}$ knows the master key MSK. When $\mathcal{A}$ generates the k-th key request query of the associated attribute set, $\mathcal{B}$ randomly selects exponent $\alpha, a, k \in \mathbb{Z}_N$ and random elements $R, R', R'', \{R_i\} \in G_{p_3}$, then calculates the following components (see the formula (6)):

We note that the generated key is a semifunctional key. In response to the first $k-1$ key query of a, $\mathcal{B}$ generates a semifunctional key according to the following. First, the normal key generation algorithm is called to generate the normal key $K, K', K'', \{K_i\}_{i\in\chi_S}$, and then, the random value τ, τ' is selected to generate the semifunctional key: $K(Y_2Y_3)^\tau, K'(Y_2Y_3)^{\tau'}, K'', \{K_i\}_{i\in\chi_S}$, where group elements $Y_2^\tau, Y_2^{\tau'}$ are distributed uniformly and randomly in G_{p_2}.

To generate a semifunctional challenge ciphertext, $\mathcal{B}$ randomly selects a vector $\tilde{v} \in Z_N^n$, and the first element of the vector has a value of 1, while letting$v = s\tilde{v}$, randomly selects exponent $\tilde{r}_x \in \mathbb{Z}_N, x \in [1, \ell]$, and sets $r_x = s\tilde{r}_x$. We should note that the elements s, v, r_x are distributed randomly; then, the corresponding ciphertext is

$$C = g_1^s Y_2, C' = m_b \cdot T, C'' = (g_1^s Y_2)^k, C_{1,x} = (g_1^s Y_2)^{a\cdot M_x\cdot\tilde{v}\cdot(-\tilde{r}_x)\cdot\eta_{\rho(x)}}, D_{1,x} = (g_1^s Y_2)^{\tilde{r}_x}. \quad (7)$$

This ciphertext is a semifunctional ciphertext, where $g_2^{s'}$ is equivalent to Y_2, a' is equivalent to $a \mod p_2$, ω is equivalent to $s\tilde{v} \mod p_2$, $\eta_{\rho(x)}$ is equivalent to$t_{\rho(x)} \mod p_2$, and $g_2^{r_x} = X_2^{\tilde{r}_x}$. These values are randomly distributed, because Y_2 are random elements in G_{p_2}, and the value of the k, $s_i, \tilde{r}_x, \tilde{v} \mod p_2$ distributed is independent of the value of these elements $\mod p_1$.

If $T = e(g_1, g_1)^{\alpha s}$, the generated ciphertext is a semifunctional ciphertext by encrypting m_b, and $\mathcal{B}$ simulates the game $Game_Q$ and interacts with $\mathcal{A}$. If $T$is a random element in G_T, then it is a semifunctional ciphertext generated by encrypting a random message, $\mathcal{B}$ simulation game $Game_{final}$. Therefore, $\mathcal{B}$ can take advantage of $\mathcal{A}$'s nonnegligible difference in these games to obtain a nonnegligible advantage to break Assumption 3.

This completes proof of Theorem 5.

7. Performance Analysis

The proposed scheme is compared with the schemes [9, 11, 14, 22–24] from the perspectives of function and computing cost. In the comparison, G_{p_i} represents the subgroup of the order p_i. N indicates the number of attribute universe. $|\ell|$ represents the number of the matrix $\boldsymbol{M}$ row, and $|y|$ represents the number of attribute sets that satisfy the policy. We use E_G,E_{G_T}, and P to denote 1 module exponential time executed in G, a module exponential time executed in G_T, and 1 bilinear pair time executed, respectively. Because the main computing overhead of this scheme contains linear pairwise operation and modular exponentiation, module multiplication and hash operation can be ignored.

7.1. Theoretical Analysis. Table 1 mainly shows the comparison of the functionality of the scheme. It can be seen that Zhang et al. [9], Lewko and Waters [14], Wang et al. [24], and our scheme are constructed on the composite order group, and these schemes are proved to be fully secure, while the schemes of Li et al. [11], Waters et al. [22], and Liu et al. [23] do not achieve full security. In terms of attribute privacy protection, besides our scheme, Zhang et al. [9] and Li et al. [11] also implemented partial policy hidden. In terms of reducing computational overhead, Li et al. [11] and Waters et al. [22] adopt offline/online technology to solve the problem, while Liu et al. [23] and Wang et al. [24] not only adopt offline/online technology but also support outsourced decryption algorithms. However, the scheme of Liu et al. [23] supports the verification of outsourced decryption results, while the scheme of Wang et al. [24] does not implement verification mechanism. Based on the above analysis, the scheme proposed in this paper not only realizes the information hiding of attribute values but also adopts offline/online technology and verifiable outsourced decryption algorithms to reduce the user's local computational cost. Besides, it is proven to be fully secure.

Table 2 gives the analysis from the computing cost. Since the literatures [14, 22, 23] do not support the policy hidden function, no analysis and comparison are listed in Table 2.

Table 1: Comparison of performance.

Scheme	Policy hidden	Composite order	Fully secure	Offline/online key generation	Offline/online encryption	Outsourced decryption	Verify
Zhang et al. [9]	✓	✓	✓	✗	✗	✗	✗
Li [11]	✓	✗	✗	✓	✓	✗	✗
Lewko and Waters [14]	✗	✓	✓	✗	✗	✗	✗
Waters [22]	✗	✗	✗	✓	✓	✗	✗
Liu et al. [23]	✗	✗	✗	✓	✓	✓	✓
Wang et al. [24]	✗	✓	✓	✓	✓	✓	✗
Ours	✓	✓	✓	✓	✓	✓	✓

Table 2: Comparison of computing overhead.

Scheme	Offline encryption	Online encryption	Outsourced decryption	Local decryption
Zhang et al. [9]	—	$(6\|\ell\|+2)E_G + 2E_{G_T}$	—	$(2\|y\|+3)P + \|y\|E_G + \|y\|E_{G_T}$
Li [11]	$(4\|\ell\|+1)E_G + 1E_{G_T}$	$1E_G$	—	$(3\|y\|+2)P + (\|y\|+1)E_G + \|y\|E_{G_T}$
Wang et al. [24]	$(3\|\ell\|+4)E_G + 1E_{G_T}$	0	$(3\|y\|+2)P + \|y\|E_G + \|y\|E_{G_T}$	$1E_G + 2E_{G_T}$
Ours	$(4\|\ell\|+3)E_G + 1E_{G_T}$	$2E_G$	$(2\|y\|+2)P + \|y\|E_{G_T}$	$2E_G + 1E_{G_T}$

It can be seen that the amount of computing required in the data encryption and data decryption stages is linearly and positively related to the number of attributes. The literatures [11, 24] and the proposed scheme use offline/online key generation and offline/online encryption technology. Therefore, most of the computing overhead in the data encryption process are performed in the offline phase, while it requires only a small amount of computing cost to complete key generation and data encryption operation in the online phase. In the scheme of [9], the modular exponentiation operation in the encryption process is much higher than other schemes. The literature [11] and our scheme have roughly the same modular exponentiation time. Although, the encryption cost of literature [24] is less than the $|\ell|$ modular exponentiation operation in literature [11] and our scheme. In the decryption process, our scheme is less than the $|y|$ linear pair operation and $|y|$ modular exponentiation operation in literature [24]. Compared with the scheme [11], our scheme is less than the $|y|$ linear pair operation and $(|y|+1)$ modular exponentiation operation. Compared with the scheme [9], our scheme is less than the $|y|$ modular exponentiation operations. Therefore, the computational efficiency of our scheme is better than other related schemes.

7.2. Experiment Analysis. Through the above theoretical analysis, the proposed scheme has more advantages in term of function and efficiency. In order to evaluate the actual performance more accurately, we perform the experiment analysis. Because the literature [11] is based on prime order groups, and other schemes are based on composite order groups, for better comparison, we only analyze the time spent in literature [9] and literature [24] through simulation experiments, including the time required of offline encryption, online encryption, outsourced decryption, and local decryption.

Experimental environment: Windows 10, Inter® Core(TM) i5-8300H (2.30 GHz), memory 8GB, the experimental code is based on JPBC-2.0.0 (Java Pairing-Based Cryptography Library) function library 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 CP-ABE, the number of attributes in the access policy affects the encryption and decryption time. In the experiment process, we set the number of attributes as 20 and increase by 5 number of attributes each time, so it is tested with 4 different access policies. By comparing the computing time of the terminal user under different access policies, we can obtain the required time.

Figure 1 has four subfigures, Figures 1(a)–1(d), which represent the data owner's offline encryption, online encryption time, cloud server decryption time required for outsourced partial decryption, and local user's decryption time.

We can see in Figure 1(a) that the offline encryption time of our scheme is higher than the time of the literature [24]. In Figure 1(b), the online encryption process of the literature [24] does not involve modular exponentiation and pairing operation, so the computing time is 0, but compared with the literature [9], the encryption time of our scheme is constant, and its time of consumption is much lower than that of the literature [9]. In Figure 1(c), the decryption overhead performed by the cloud server is lower than that in the literature [24]. In Figure 1(d), the local decryption time of our scheme and that of the literature [24] are both constant, which is much lower than that of the scheme of literature

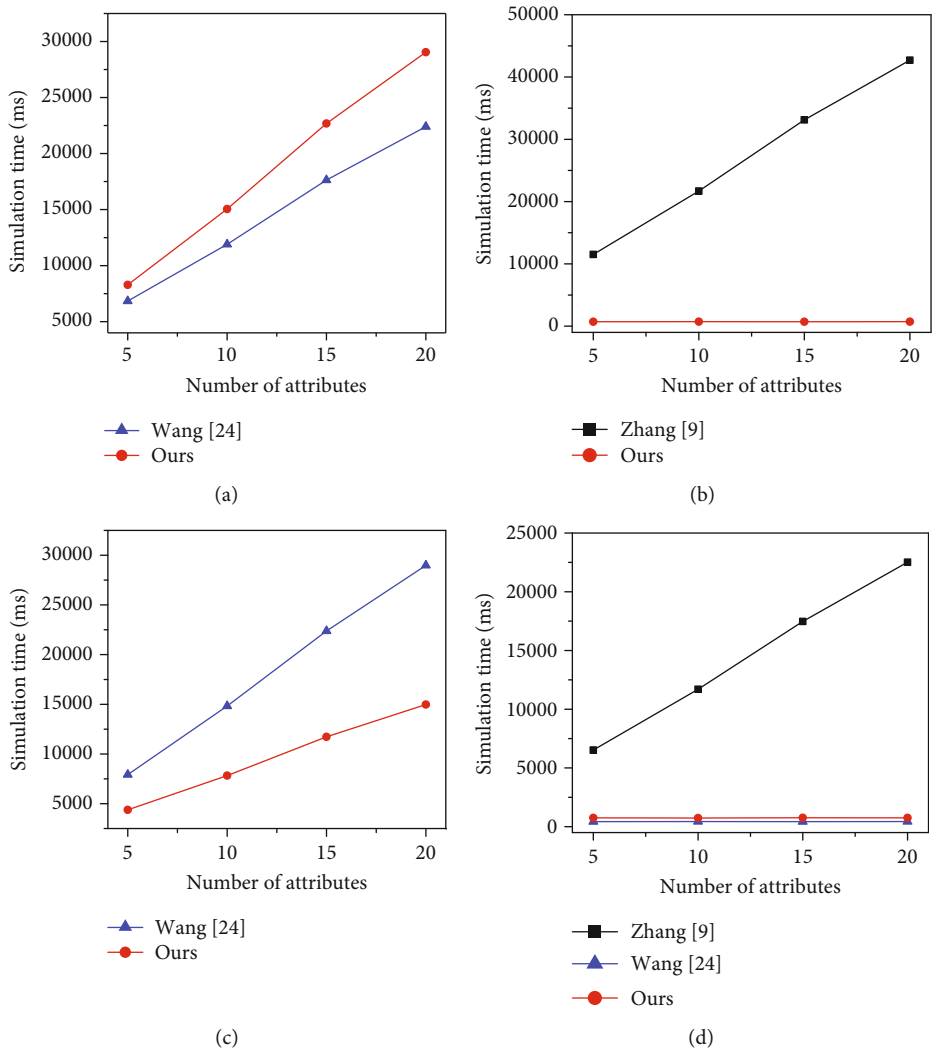

Figure 1: Simulation experiment. (a) Offline encryption. (b) Online encryption. (c) Outsourced decryption. (d) Local decryption.

[9]. The decryption time required is slightly higher than scheme of [24], but in [24], the partial decryption ciphertext returned by the cloud server is not supported verification; then, the correctness is not guaranteed. Meanwhile, Wang et al.'s scheme [24] cannot realize the policy hiding function. Since the proposed scheme supports outsourced decryption operations and verification operations, the user only needs to perform a constant number of exponential operations, which can not only reduce the user's calculation burden but also ensure the accuracy of partial decryption result returned.

From the above comprehensive analysis, the proposed scheme is superior to other schemes in terms of function and performance, so it is more effective and feasible in the IoT environment.

8. Conclusion

In order to solve the problem of privacy leaking and heavy computing overhead in IoT environment, an offline/online outsourced ABE scheme with partial policy hidden is proposed in the paper. In the scheme, it divides attributes into two parts: attribute name and attribute value, attribute name is open and attribute value is hidden, to achieve the privacy of user attributes. Additionally, the offline/online technology is adopted to reduce the burden of encryption and decryption. A lot of heavy work can be preprocessed in the offline stage, and the rest computation only need to be done in the online stage. For the bilinear pairing operation and module power operation, the operation will be outsourced to the cloud server, and the user only needs to verify the outsourced calculation results to ensure the accuracy. It is proven that the scheme based on the static assumption problem can achieve full security under the standard model. Lastly, through theoretical and experiment analysis, it shows that our scheme has more advantages in the IoT environment.

Data Availability

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

Conflicts of Interest

The authors declare there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported in part by the Special Focus on Research and Promotion of Henan Province under Grant 192102210280, in part by the Research Foundation of Young Core Instructor in Henan Province under Grants 2018GGJS058 and 2019GGJS061, and in part by the Innovative Scientists and Technicians Team of Henan Provincial High Education under Grant 20IRTSTHN013.

References

[1] A. Sahai and B. Waters, "Fuzzy identity-based encryption," *Proceedings of the Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 457–473, Aarhus, Denmark, 2005.

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

[3] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," *Proceedings of the IEEE Symposium on Security and Privacy*, pp. 321–334, Berkeley, USA, 2007.

[4] T. Nishide, K. Yoneyama, and K. Ohta, "Attribute-based encryption with partially hidden encryptor-specified access structures," in *Proceeding Application Cryptography Network Security(ACNS)*, pp. 13–23, Springer, 2009.

[5] J. Lai, R. H. Deng, and Y. Li, "Expresssive CP-ABE with partially hidden access structures," *Proceedings of the 7th ACM Symposium on Information, Computer and Communications Security*, pp. 18-19, Seoul, South Korea, 2012.

[6] J. Li, H. Wang, Y. Zhang, and J. Shen, "Ciphertext-policy attribute-based encryption with hidden access policy and testing," *KSII Transactions on Internet and Information Systems*, vol. 10, no. 7, pp. 3339–3352, 2016.

[7] H. Yin, L. Zhang, and Y. Cui, "Improving security in ciphertext-policy attributed-based encryption with hidden access policy and testing," *KSII Transactions on Internet and Information Systems*, vol. 10, no. 7, pp. 3339–3352, 2016.

[8] H. Cui, R. H. Deng, J. Lai, X. Yi, and S. Nepal, "An efficient and expressive ciphertext-policy attribute-based encryption scheme with partially hidden access structures, revisited," *Computer Networks*, vol. 133, pp. 157–165, 2018.

[9] Y. Zhang, D. Zheng, and R. H. Deng, "Security and privacy in smart health: efficient policy-hiding attributed-based access control," *IEEE internet of thing journal*, vol. 5, no. 3, pp. 1–15, 2018.

[10] L. Zhang, G. Hu, Y. Mu, and F. Rezaeibagha, "Hidden ciphertext policy Attribute-Based encryption with fast decryption for personal health record system," *IEEE Access*, vol. 7, pp. 33202–33213, 2019.

[11] X. Li, H. Tian, and J. Ning, "Secure online/offline attribute-based encryption for IoT users in cloud computing," *International Conference on Provable Security*, pp. 347–354, 2019.

[12] B. Waters, "Dual system encryption: realizing fully secure IBE and HIBE under simple assumption," in *Annual International Cryptology Conference*, pp. 619–636, Santa Barbara,CA,USA, 2009.

[13] A. Lewko, T. Okamoto, A. Sahai, K. Takashima, and B. Waters, "Fully secure functional encryption: attribute-based encryption and (hierarchical)inner product encryption," *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 62–91, 2010.

[14] A. B. Lewko and B. Waters, "New proof methods for attribute-based encryption: achieving full security through selective techniques," *Annual Cryptology Conference*, 2012, pp. 180–198, Santa Barbara, CA, USA, August 2012.

[15] C. Jin, X. Feng, and Q. Shen, "Fully secure hidden ciphertext policy attribute-based encryption with short ciphertext size," *Proceedings of the 6th International Conference on Communication and Network Security*, pp. 91–98, Singapore, 2016.

[16] Y. Zhang, X. Chen, J. Li, D. S. Wong, H. Li, and I. You, "Ensuring attribute privacy protection and fast decryption for outsourced data security in mobile cloud computing," *Information Sciences*, vol. 379, pp. 42–61, 2017.

[17] X. Yan, H. Ni, Y. Liu, and D. Han, "Privacy-preserving multi-authority attribute-based encryption with dynamic policy updating in PHR," *Computer Science and Information Systems*, vol. 16, no. 3, pp. 831–847, 2019.

[18] A. Lewko, A. Sahai, and B. Waters, "Predicate encryption supporting disjunctions, polynomial equations, and inner products," *Annual international conference on the theory and applications of cryptographic techniques*, pp. 146–162, 2008.

[19] S. Even, O. Goldreich, and S. Micali, "Online/offline digital signatures," *Proceedings of the Conference on the Theory and Application of Cryptology*, pp. 263–275, Santa Barbara, USA, 1989.

[20] J. K. Liu, J. Baek, J. Zhou, Y. Yang, and J. W. Wong, "Efficient online/offline identity-based signature for wireless sensor network," *International Journal of Information Security*, vol. 9, no. 4, pp. 287–296, 2010.

[21] F. C. Guo, Y. Mu, and Z. D. Chen, "Identity-based online/offline encryption," *Proceedings of the International Conference on Financial Cryptography and Data Security*, pp. 247–261, Cozumel, Mexico, 2008.

[22] S. Hohenberger and B. Waters, "Online/offline attribute-based encryption," *Proceedings of the International Workshop on Public Key Cryptography*, pp. 293–310, Buenos Aires, Argentina, 2014.

[23] Z. Liu, Z. L. Jiang, X. Wang, X. Huang, S. M. Yiu, and K. Sadakane, "Offline/online attribute-based encryption with verifiable outsourced decryption," *Concurrency and Computation: Practice and Experience*, vol. 29, no. 7, pp. 1–17, 2017.

[24] H. Wang, Z. Zheng, and Y. Wang, "Cloud-aided online/offline ciphertext-policy attribute based encryption in the standard model," *International Journal of Grid and Utility Computing*, vol. 8, no. 3, pp. 211–221, 2017.

[25] W. Liang, K. C. Li, J. Long, X. Kui, and A. Y. Zomaya, "An industrial network intrusion detection algorithm based on multifeature data clustering optimization model," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 3, pp. 2063–2071, 2020.

[26] W. Liang, W. Huang, J. Long, K. Zhang, K. C. Li, and D. Zhang, "Deep reinforcement learning for resource protection and real-time detection in IoT environment," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 6392–6401, 2020.

[27] M. Green, S. Hohenberger, and B. Waters, "Outsourcing the decryption of ABE ciphertexts," *Proceedings of the 20th USENIX Conference on Security*, pp. 1–16, San Francisco, USA, 2011.

[28] J. Lai, R. H. Deng, C. Guan, and J. Weng, "Attribute-based encryption with verifiable outsourced decryption," *IEEE Transactions Information Forensics and Security*, vol. 8, no. 8, pp. 1343–1354, 2013.

[29] J. Li, Y. Zhang, X. Chen, and Y. Xiang, "Secure attribute-based data sharing for resource-limited users in cloud computing," *Computers & Security*, vol. 72, pp. 1–12, 2018.

[30] K. Fan, J. Wang, X. Wang, H. Li, and Y. Yang, "A secure and verifiable outsourced access control scheme in fog-cloud computing," *Sensors*, vol. 17, no. 7, pp. 1695–1710, 2017.

[31] R. Zhang, H. Ma, and Y. Lu, "Fine-grained access control system based on fully outsourced attribute-based encryption," *Journal of Systems and Software*, vol. 125, no. 3, pp. 344–353, 2017.

[32] Z. Y. Zhao, J. H. Wang, K. Y. Xu, and S. H. Guo, "Fully outsourced attribute-based encryption with verifiability for cloud storage," *Journal of Computer Research and Development*, vol. 56, no. 2, pp. 442–452, 2018.

[33] J. Yu, G. He, X. Yan, Y. Tang, and R. Qin, "Outsourced ciphertext-policy attribute-based encryption with partial policy hidden," *International Journal of Distributed Sensor Networks*, vol. 16, no. 5, pp. 1–14, 2020.

[34] W. Liang, Y. Fan, K. C. Li, D. Zhang, and J. L. Gaudiot, "Secure data storage and recovery in industrial blockchain network environments," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 10, pp. 1–6552, 2020.