CR-FH-CPABE: Secure File Hierarchy Attribute-Based Encryption Scheme Supporting User Collusion Resistance in Cloud Computing

Yuhan Bai, Kai Fan, *Member, IEEE*, Kuan Zhang, *Member, IEEE*, Hui Li, *Senior Member, IEEE*, and Yintang Yang, *Senior Member, IEEE*

***Abstract*—The attribute-based encryption (ABE) scheme, which can set specific conditions to control user access to data, has been widely studied and applied to cloud storage services. Considering file hierarchy in practical scenarios, the ABE scheme can set a hierarchical access control policy so multiple files can be associated with one access structure to reduce users' computing overhead and save the cloud server's storage space. However, the existing systems have the risk of user collusion due to the hierarchical access control structure parameters. This article proposes a secure file hierarchy ABE scheme supporting user collusion resistance (CR-FH-CPABE) in cloud computing. We add a data noise vector without changing the hierarchical access control structure to prevent user ultra vires. Technically, we break the relationships that colluding users could exploit, prevent malicious users from colluding with their computing results, and extract meaningful information from the ciphertext. In addition, we provide an improved CR-FH-CPABE scheme with outsourced decryption, which helps resource-limited devices obtain computing services. Finally, we demonstrate our scheme is CPA secure and show outstanding performance through simulation results.**

***Index Terms*—Attribute-based encryption (ABE), file hierarchy, hierarchical access control, outsourced decryption, user collusion resistance.**

Manuscript received 2 December 2023; revised 11 January 2024; accepted 23 January 2024. Date of publication 29 January 2024; date of current version 9 May 2024. This work was supported in part by the "Pioneer" and "Leading Goose" Research and Development Program of Zhejiang under Grant 2022C03174; in part by the National Natural Science Foundation of China under Grant 62372356 and Grant 92067103; in part by the Key Research and Development Program of Shaanxi under Grant 2021ZDLGY06-02; in part by the Henan Key Laboratory of Network Cryptography Technology under Grant LNCT2022-A13; in part by the Natural Science Foundation of Shaanxi Province under Grant 2019ZDLGY1202; in part by the Shaanxi Innovation Team Project under Grant 2018TD-007; in part by the Xi'an Science and Technology Innovation Plan under Grant 20189168CX9JC10; in part by the Fundamental Research Funds for the Central Universities under Grant YJS2212; and in part by the National 111 Program of China under Grant B16037. *(Corresponding author: Kai Fan.)*

Yuhan Bai and Kai Fan are with the State Key Laboratory of Integrated Service Networks, School of Cyber Engineering, Xidian University, Xi'an 710126, Shaanxi, China, and also with the Henan Key Laboratory of Network Cryptography Technology, Zhengzhou 450003, China (e-mail: yhbai@stu.xidian.edu.cn; kfan@mail.xidian.edu.cn).

Kuan Zhang is with the Department of Electrical and Computer Engineering, University of Nebraska–Lincoln, Lincoln, NE 68588 USA (e-mail: kuan.zhang@unl.edu).

Hui Li is with the State Key Laboratory of Integrated Service Networks, School of Cyber Engineering, Xidian University, Xi'an 710071, Shaanxi, China (e-mail: lihui@mail.xidian.edu.cn).

Yintang Yang is with the Key Laboratory of the Ministry of Education for Wide Band Gap Semiconductor Materials and Devices, Xidian University, Xi'an 710071, China (e-mail: ytyang@xidian.edu.cn).

Digital Object Identifier 10.1109/JIOT.2024.3358745

I. Introduction

As an attractive computing paradigm, cloud computing has become an essential part of the digital transformation for many organizations [1]. This is primarily due to the popularity of the Internet, which makes it easier for people to access cloud computing resources, and the reduction in computing costs makes cloud computing more affordable for enterprises. However, as more and more companies shift their business to the cloud, maintaining data privacy during the storage and access process remains a significant challenge [2]. Additionally, file hierarchy is vital in specific scenarios associated with files [3]. To provide a better understanding, we give the following two file hierarchy examples [4], [5], [6].

1) *Team Collaboration:* In many workplaces, teams often must collaborate on the same files to complete projects and tasks. Each team member may have different access rights and responsibilities for processing specific file parts. In such cases, file hierarchy in cloud storage can facilitate team members' access to different parts of the same file from various locations and devices. This enables seamless collaboration and improves overall productivity.
2) *Healthcare:* Accessing and sharing patient data is crucial for optimal care in the healthcare industry. However, patient data contains personal and medical information, including sensitive details unrelated to the treatment process. To maintain data privacy, healthcare professionals must ensure they only access and share the necessary data for collaboration. File hierarchy in cloud storage can help organize and manage patient data, enabling healthcare teams to collaborate efficiently and legally.

While file hierarchy in cloud computing offers advantages such as promoting collaboration and improving work efficiency and decision-making quality, it is important to address potential issues like user collusion. User collusion refers to the unauthorized collaboration or sharing of sensitive data among users not authorized to access or manipulate specific data. In a cloud environment where multiple users have access to shared files with file hierarchy, it is essential to implement appropriate security measures to mitigate the risk of user collusion.

Ciphertext-policy attribute-based encryption (CP-ABE) is a promising technology to set data access solutions and storage ways securely in cloud computing, which can meet

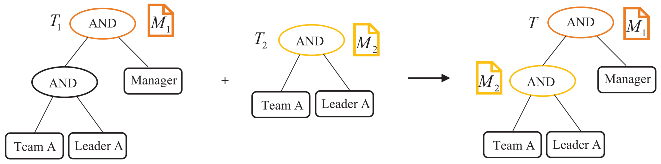


Fig. 1. Relationship of access structures in the file hierarchy.

the flexibility requirements of cloud storage [7]. The original CP-ABE scheme [8] allows users to use one attribute set to set the specific access policy and enables multiple users who meet the attribute set to access the same file. However, CP-ABE has limitations, such as complex key management and distribution processes and high computation overhead. To address these challenges, a CP-ABE scheme with file hierarchy (FH-CP-ABE) [9] was proposed for organizing files in a hierarchical structure to reduce storage and computation costs associated with CP-ABE. It allows for more efficient management and access control of files with file hierarchy while still maintaining the security and flexibility offered by CP-ABE. By using this kind of hierarchical structure, it can further meet the needs of multiuser shared decryption in practical applications [10].

A. Motivation

Existing research has achieved hierarchical access to files with a hierarchy access tree. Here, we provide one example to introduce the relationship of access structures in the file hierarchy. We set a manager {*Manager*} to be responsible for the management of a team {*Team A*}, and this team {*Team A*} has a corresponding leader {*Leader A*}. As shown in Fig. 1, we can know the manager with an attribute set {*Manager, Team A, Leader A*} can recover the file M_1 and the leader with an attribute set {*Team A, Leader A*} can recover the file M_2. It is worth noting that the manager can also recover file M_2 with $\{Team\ A, Leader\ A\} \subset \{Manager, Team\ A, Leader\ A\}$. This inclusion relationship $\mathcal{T}_2 \subset \mathcal{T}_1$ allows for the merging of access structures into an integrated access structure $\mathcal{T}$ to establish the file hierarchy.

While hierarchical access control facilitates different access levels and simplifies management, it is essential to recognize that it cannot prevent user collusion. Merging access control structures of multiple files into one can optimize storage and management. However, it can also introduce vulnerabilities. For example, suppose two files with unrelated access control settings are merged into a third file. In that case, malicious users who individually have access to the original two files can collude to gain access to the third file. This can lead to unauthorized exposure of confidential information. To address this challenge, it is vital to carefully consider the implications of using hierarchical access control policies and design an appropriate encryption algorithm, which should resist user collusion and effectively safeguard the confidentiality of sensitive data.

B. Our Contribution

We have proposed the secure file hierarchy attribute-based encryption scheme supporting user collusion resistance (CR-FH-CPABE) to address the limitations of existing FH-CP-ABE schemes and provide resistance against user collusion. Our contributions include the following five points.

1) *Data Noise Vector:* We use data noise vector in our CR-FH-CPABE scheme. Data noise vector refers to a sequence of randomly generated meaningless values added to the ciphertext, which aims to effectively resist user collusion by obscuring the secret values in the ciphertext.
2) *File Hierarchy With User Collusion Resistance:* Adding data noise in the form of random values to the ciphertext can preserve the overall structure and hierarchical access policy of the original ciphertext. The noise disrupts the relationships that colluding users could exploit, which makes it difficult for malicious users to collude their computing results and extract meaningful information from the ciphertext.
3) *Flexible Decryption:* We provide the improved CR-FH-CPABE scheme, which can support outsourced decryption. Users can delegate their keys to the server to obtain outsourced computing, which helps resource-limited devices obtain computing services.
4) *Enhanced Security:* We demonstrate the security of our proposed CR-FH-CPABE scheme, which is chosen plaintext attacks (CPA) secure.
5) *Performance and Efficiency:* We conducted simulations using Java language to demonstrate our practicality, then compared and analyzed them with existing works. The simulation results show the outstanding advantages of our solution.

C. Organization

The remaining part of this article is organized as follows. In Section II, we review the existing work. Section III introduces the technical preliminaries. In Section IV, we give the system model and security assumptions for problem definition. In Section V, we introduced our proposed CR-FH-CPABE scheme and improved CR-FH-CPABE scheme. Sections VI and VII offer our security proof and performance evaluation. Finally, Section VIII is a summary of this article.

II. Related Work

A. Taxonomy of CP-ABE

As shown in Fig. 2, we present a taxonomy of existing CP-ABE schemes and provide their further details on each category.

1) *Access Structure:* Research on access control structures can be categorized into three types: a) AND-gate structures [11]; b) tree structures [12]; and c) linear secret sharing schemes (LSSS) [13]. CP-ABE schemes that support AND gates and threshold policies offer complex access control capabilities but increase ciphertext length and computation overhead. Tree structures, with a hierarchical relationship, provide high expressiveness and scalability. LSSS uses matrices to enhance computational efficiency, but may lack flexibility in extending policies.

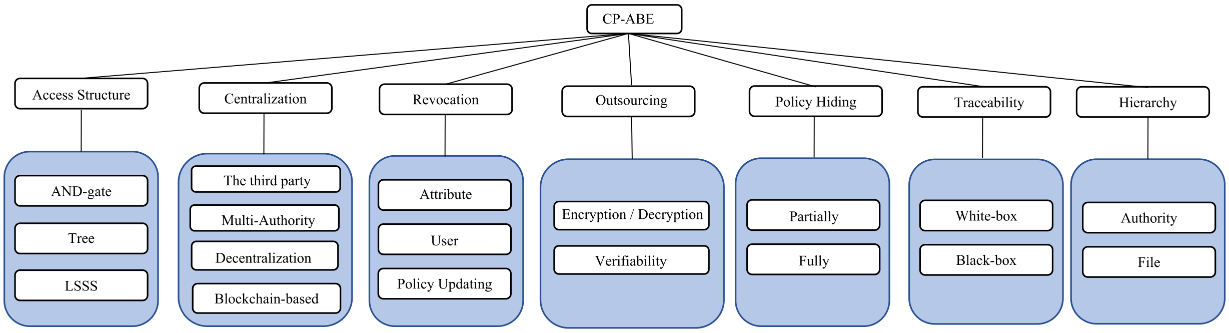


Fig. 2. Taxonomy of CP-ABE.

2) *Centralization:* This category focuses on maintaining the trustworthiness of the third party in CP-ABE schemes while improving system scalability, security, and efficiency. Although centralized CP-ABE systems are easy to implement and manage, they are susceptible to attacks and present a risk of single-point failure [12]. To address these issues, alternative approaches, such as multiauthority [14], [15], decentralization [16], and blockchain-based [17] schemes, have been proposed to distribute trust.
3) *Revocation:* Revocation involves denying access to users who leave the organization or when access control policies change [18]. As the system evolves, users, attributes, and access policies may dynamically change. Therefore, revocation mechanisms must ensure forward security, backward security and data integrity while considering efficiency [19], [20], [21].
4) *Outsourcing:* Outsourcing refers to delegating encryption and decryption tasks to a proxy server, often provided by a cloud service provider. Users can offload computationally intensive tasks to proxy servers to enhance computing efficiency [22]. To ensure the correctness of outsourcing results, effective verification mechanisms are required to validate computations performed by the proxy server [23].
5) *Policy Hiding:* Access policies are generated and stored in the cloud, which means the cloud service provider has access to the access structure and could potentially reveal sensitive information about the data and users. Policy hiding techniques provide additional privacy protection for access structures and can be classified into two types: a) partial policy hiding [24] and b) full policy hiding [25].
6) *Traceability:* Traceability is employed to track users who may have leaked or tampered with encryption keys, holding them accountable. This can be achieved by incorporating traceable components. Depending on the attackers' capabilities, traceability mechanisms can be classified as white box [26] and black box [27].

B. Hierarchical CP-ABE

Gentry and Silverberg [28] proposed a hierarchical scheme with identity-based encryption that allows higher level private key generators to delegate management to lower level ones, reducing the burden of distributing keys by authorities. Since attribute-based encryption (ABE) was proposed, Li et al. [29] have set an attribute hierarchy ABE scheme based on attribute classification. However, this scheme only establishes access structures with attribute paths. Based on CP-ABE, Deng et al. [30] defined a hierarchical ABE (HABE) scheme. But it only focuses on the layered delegation of trusted third-party keys and does not consider the relationships between access control policies. Therefore, early research focused more on the hierarchical refinement of key distribution centers to achieve structured key generation and distribution work without considering the connections between data [31], [32].

In practical scenarios, the access control policies set by the same user for a group of personal data usually have a hierarchical structure, such as the access structure of one file is a subset of the access structure of another file. If these data are encrypted using the same hierarchical access structure, multiple files can be effectively compressed, thereby saving ciphertext storage and encryption costs. Wang et al. [9] first proposed an FH-CP-ABE scheme that encrypts multiple files using an integrated access structure. The associated attribute ciphertexts of the integrated access control tree only need to be calculated once, effectively saving computation and storage costs. This scheme also sets up transmission ciphertexts to improve decryption efficiency, but its calculation method for transmission ciphertexts can recover messages that users cannot access, making it unable to achieve CPA security. Subsequently, schemes [3], [4], and [33] proposed revocation and multiauthority schemes based on Wang et al. [9], and they are not secure as they all used the same form of transmission ciphertexts.

Li et al. [34] modified the form of transmitted ciphertexts and solved the way of calculating between transmitted ciphertexts but ignored the problem of collusion among multiple users. Based on this ciphertext form, multiple users who decrypt subtree access structures can collude to decrypt the root node ciphertext. Subsequently, attribute-based hierarchical schemes with revocation [35] and hash-based file access schemes [36] appeared, which proposed higher functional requirements for the above hierarchical access control schemes. These schemes [35], [36], [37], [38] did not use the transmission ciphertexts, thus achieving CPA security for encryption algorithms. However, these schemes ignore resisting user collusion attacks.

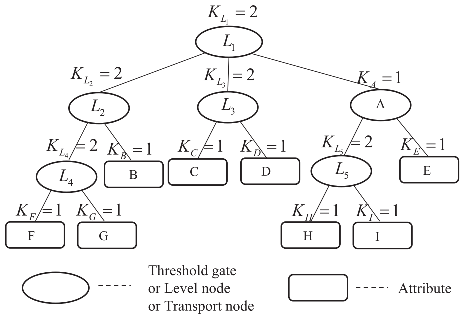


Fig. 3. Access tree.

TABLE I
ADDITIONAL FUNCTIONS

Notation	Meaning
num_x	Child number of node x in $\mathcal{T}$
$\mathcal{T}_x$	Tree diagram with x as the root node
k_x	Threshold value of node x in $\mathcal{T}$ with $0 < k_x \leq num_x$
$parent\,(x)$	Parent number of node x in $\mathcal{T}$
$att\,(x)$	Corresponding attribute of the leaf node x in $\mathcal{T}$
transport node	Children of this node contain at least one threshold gate
$TN - CT\,(x)$	Threshold gate set of child node of transport node x
$index\,(x)$	Return the corresponding leaf node x value

III. PRELIMINARIES

We use bilinear maps [12], secret sharing scheme [38], and access tree [34] to construct our schemes.

A. Secret Sharing Scheme

Choose a polynomial function $q(x) = a_0 + a_1 x + \cdots + a_{t-1}x^{t-1}\ mod\ p$, where p is a prime and $a_0 = s \in \mathbb{Z}_p$ is the secret value. Calculate $y_i = q(x_i)$ $(1 \leq i \leq n)$ with n different numbers $x_i \in \mathbb{Z}_p$, and share y_i to n participants. When the number of participants is more than t, they can reconstruct s through the Lagrange coefficient $\Delta_{i,S}(x) = \prod_{j \neq i, j \in S} (x - x_j / x_i - x_j)$

$$s = q(0) = \sum_i y_i \Delta_{i,S}(0).$$

B. Access Tree

$\mathcal{T}$ is an access tree with threshold gates and attributes. Attributes contained in the access control structure are associated with the leaf nodes, and threshold gates are at the nonleaf nodes, such as *AND*, *OR*, and n of m $(n < m)$.

For the extended hierarchy access tree of scheme [34], if the nonleaf node has corresponding file M_i $(i = 1 \ldots \kappa)$, it is called level node $L_i \in L = \{L_1, L_2, \ldots, L_\kappa\}$. Therefore, their file hierarchy is arranged monotonically from top to bottom of $\mathcal{T}$ (or from left to right when having the same depth in $\mathcal{T}$). Additional functions and terms of the extended hierarchy access tree are shown in Table I. For example, in Fig. 3, $\text{num}_{L_1} = 3$. $k_{L_1} = 2$, $k_{L_2} = 2$, and $k_A = 1$. $\text{parent}(L_3) = L_1$. L_1, L_2, and A are transport nodes. $TN - CT(L_1) = \{L_2, L_3, A\}$, $TN - CT(L_2) = \{L_4\}$, and $TN - CT(A) = \{L_5\}$.

TABLE II
NOTATIONS SUMMARY

Notation	Meaning
λ	Security parameter
g	Generator
p	Prime order
G_0, G_T	Cyclic multiplicative groups
e	Bilinear mapping
H_i	Hash function
MPK	Master public key
MSK	Master secret key
SK_{u_k}	Private Keys of user u_k
$SK_{u_k}^{out}$	Private Keys in outsourced decryption
$TK_{u_k}^{out}$	Translation Keys in outsourced decryption
M_i	Message
ck	Symmetric key
$\mathcal{T}$	Access tree
CT	Ciphertext
uid	User identification
θ	Value of file hierarchy
S_i	Attribute set
s_i	Secret value
ε	Data noise vector

IV. PROBLEM DEFINITION

A. System Model

The system model of CR-FH-CPABE consists of four entities, which are center authority (CA), cloud server (CS), data owner (DO), and data user (DU), as shown in Fig. 4.

1) CA sets up the system and calculates the user's keys with user identification *uid*.
2) CS stores ciphertext for DO and shares the file with legal DUs. However, only DUs that meet the access control policy can recover messages.
3) DO needs to encrypt the message and store it in CS. For file hierarchy, DO must set different access levels for different file contents.
4) DU downloads the file from CS and attempts to recover the message. DU can recover the message when the attributes of DU meet the access tree.

Definition 1: Our proposed CR-FH-CPABE scheme has the following four fundamental algorithms.

1) *Setup*(λ) → (MPK, MSK) inputs a security parameter λ, then returns MPK and *MSK*.
2) *Keygen*(MPK, MSK, S_{u_k}) → (SK_{u_k}) inputs MPK, *MSK*, and the attribute set S_{u_k} of user u_k, then returns user's keys SK_{u_k}.
3) *Encrypt*(MPK, ck_i, $\mathcal{T}$)→ (CT) inputs MPK, the symmetric key ck_i and an access tree $\mathcal{T}$, then returns the ciphertext *CT*.
4) *Decrypt*(MPK, *CT*, SK_{u_k})→ (ck_i) inputs MPK, *CT*, and SK_{u_k}, then returns a symmetric key ck_i.

The notations summary is shown in Table II.

B. Security Assumptions

We have established security assumptions for four entities involved in the system.

1) CA is considered fully trusted and responsible for the security of system parameters.
2) CS is honest but curious and may attempt to acquire additional information while executing the user subscription protocol.

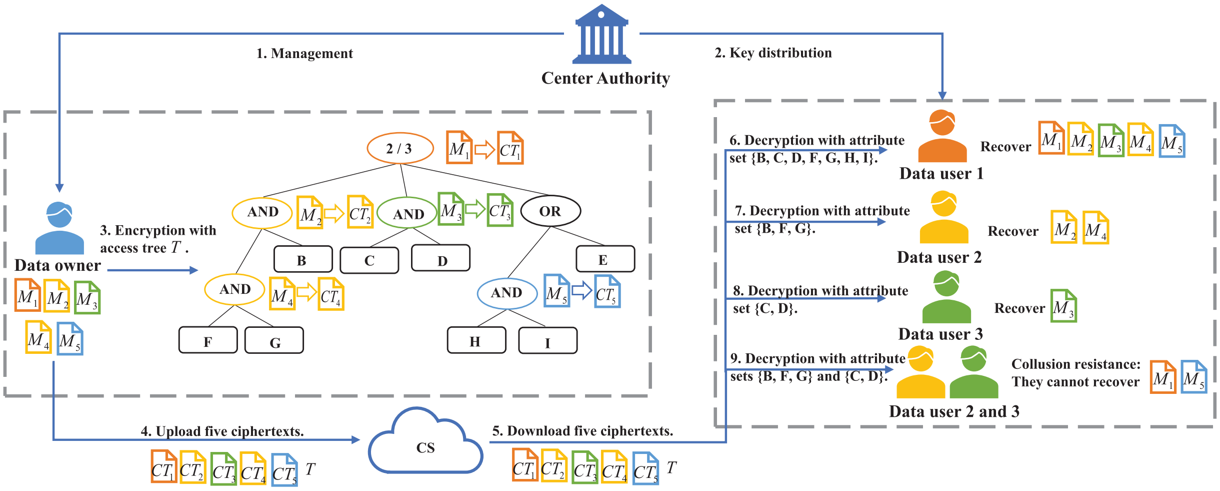

1. Management
2. Key distribution
Center Authority
Data owner
3. Encryption with access tree T.
M_1 M_2 M_3 M_4 M_5
2 / 3
AND
AND
OR
AND
AND
B
C
D
E
F
G
H
I
M_1 CT_1
M_2 CT_2
M_3 CT_3
M_4 CT_4
M_5 CT_5
6. Decryption with attribute set {B, C, D, F, G, H, I}.
Data user 1
Recover M_1 M_2 M_3 M_4 M_5
7. Decryption with attribute set {B, F, G}.
Data user 2
Recover M_2 M_4
8. Decryption with attribute set {C, D}.
Data user 3
Recover M_3
9. Decryption with attribute sets {B, F, G} and {C, D}.
Data user 2 and 3
Collusion resistance: They cannot recover M_1 M_5
4. Upload five ciphertexts.
CS
5. Download five ciphertexts.
CT_1 CT_2 CT_3 CT_4 CT_5 T
CT_1 CT_2 CT_3 CT_4 CT_5 T

Fig. 4. System model of CR-FH-CPABE: users can access files based on their attribute sets, but it is important to prevent collusion attacks among users. For instance, users DU_2 and DU_3 are not able to recover files beyond their granted permissions through collusion or by obtaining transmission ciphertexts, such as M_1 and M_5.

3) DO is fully trusted, and it is essential to protect their files under the access control policy.
4) DU is a potentially dishonest entity that may collude with others to gain access to unauthorized files. In hierarchical access scenarios, malicious DUs may attempt to collude with other users to decrypt files with higher access levels.

V. Proposed CR-FH-CPABE Scheme

A. Discussion

In file hierarchy schemes [3], [4], [9], [33], [34], [35], [36], [37], [38], it has been identified that a vulnerability exists when two child nodes are under the same parent node. In such cases, DUs accessing the files associated with these child nodes can collude to recover the file corresponding to the parent node.

For example, in Fig. 4, we suppose DU_2 can recover M_2 with $F_{L_2} = e(g,g)^{r_2 s_2}$, and DU_3 can recover M_3 with $F_{L_3} = e(g,g)^{r_3 s_3}$. Here, r_i $(i = 1, 2)$ is randomly selected by CA to distinguish the user's private keys, so they believe it is impossible to compute $F_{L_1} = e(g,g)^{r_i s_1}$ by mixing the private keys of two users in the original method

$$\begin{aligned} & F_{L_2}^{\Delta_{\text{index}(L_2),S'_{L_1}}(0)} \cdot F_{L_3}^{\Delta_{\text{index}(L_3),S'_{L_1}}(0)} \\ & = e(g,g)^{r_2 s_2 \Delta_{\text{index}(L_2),S'_{L_1}}(0) + r_3 s_3 \Delta_{\text{index}(L_3),S'_{L_1}}(0)} \\ & \neq e(g,g)^{r_i \left(s_2 \Delta_{\text{index}(L_2),S'_{L_1}}(0) + s_3 \Delta_{\text{index}(L_3),S'_{L_1}}(0) \right)} \\ & = e(g,g)^{r_i s_1} = F_{L_1} (i = 1, 2). \end{aligned}$$

However, DU_2 and DU_3 can obtain $e(g,g)^{\alpha s_2}$ and $e(g,g)^{\alpha s_3}$ during recovering M_2 and M_3, respectively. They can get the critical $e(g,g)^{\alpha s_1}$ corresponding to M_1 through user collusion

$$\begin{aligned} & \left(e(g,g)^{\alpha s_2}\right)^{\Delta_{\text{index}(L_2),S'_{L_1}}(0)} \cdot \left(e(g,g)^{\alpha s_3}\right)^{\Delta_{\text{index}(L_3),S'_{L_1}}(0)} \\ & = e(g,g)^{\alpha \left(s_2 \Delta_{\text{index}(L_2),S'_{L_1}}(0) + s_3 \Delta_{\text{index}(L_3),S'_{L_1}}(0) \right)} \\ & = e(g,g)^{\alpha s_1}. \end{aligned}$$

Therefore, malicious users corresponding to L_2 and L_3 can recover the file M_1 of the parent node L_1 and can decrypt all files (such as M_5) under L_1 by the transmission ciphertext. There is a similar problem of collusion attacks in the multiauthority FH-CP-ABE scheme [4], [33]

$$\begin{aligned} & e(g,g)^{\sum_i \alpha_i s_2 \Delta_{\text{index}(L_2),S'_{L_1}}(0)} \cdot e(g,g)^{\sum_i \alpha_i s_3 \Delta_{\text{index}(L_3),S'_{L_1}}(0)} \\ & = e(g,g)^{\sum_i \alpha_i \left(s_2 \Delta_{\text{index}(L_2),S'_{L_1}}(0) + s_3 \Delta_{\text{index}(L_3),S'_{L_1}}(0) \right)} \\ & = e(g,g)^{\sum_i \alpha_i s_1}. \end{aligned}$$

Therefore, the secret values s_i generated by the same polynomial function create computable relationships between ciphertexts CT_i, providing conditions for user collusion. In our CR-FH-CPABE scheme, we reset ciphertext by adding a data noise vector to resist collusion attacks.

B. Construction of CR-FH-CPABE Scheme

This section provides the details of the algorithms in our CR-FH-CPABE scheme.

1) Setup: Setup(λ) $\rightarrow$ *(MPK, MSK)* is run by CA for system initialization. CA inputs a security parameters λ and chooses a bilinear mapping $e : G_0 \times G_0 \rightarrow G_T$ with a cyclic multiplicative group G_0, a generator g, and a prime order p. This algorithm defines the hash functions as $H_1 : \{0,1\}^* \rightarrow G_0$, $H_2 : G_T \rightarrow \mathbb{Z}_p$ and $H_3 : \{0,1\}^* \rightarrow \mathbb{Z}_p$. Then, CA randomly selects $\alpha, \beta_1, \beta_2, \beta_3 \in \mathbb{Z}_p$, and published MPK is

$$\{g, G_0, e, H_1, H_2, f_1 = g^{\beta_1}, f_2 = g^{\beta_2}, f_3 = g^{\beta_3}, e(g,g)^{\alpha}\}.$$

Let $h_1 = g^{\beta_1^{-1}}$, $h_2 = g^{\beta_2^{-1}}$, and $h_3 = g^{\beta_3^{-1}}$. CA randomly chooses $\theta \in \mathbb{Z}_p$ as the value of file hierarchy, and MSK is

$$\{\alpha, \beta_1, \beta_2, \beta_3, \theta, H_3\}.$$

2) Key Generation: Keygen$(MPK, MSK, S_i) \rightarrow (SK_i)$ is run by CA to generate private keys for user. For the ith user with user identification uid, CA randomly chooses $r_i, \omega_1 \in \mathbb{Z}_p$ where $\omega_1 = H_3(uid)$ for user, and $r_{i,j} \in \mathbb{Z}_p$ $(j \in [1, N_i])$ for each attribute in $S_i = \{\text{att}_{i,1}, \ldots, \text{att}_{i,N_i}\}$. CA computes the private key SK_i as

$$\left\{ D_i = g^{\frac{\alpha+\theta}{\beta_1}}, E_i = g^{\frac{\theta+r_i+\omega_1}{\beta_2}}, E_i' = g^{\frac{r_i+\omega_1}{\beta_3}} \right.$$
$$\left. \forall j \in [1, N_i] : \overset{\frown}{D}_{i,j} = g^{r_i+\omega_1} H_1\left(\text{att}_{i,j}\right)^{r_{i,j}}, \overset{\smile}{D}_{i,j} = g^{r_{i,j}} \right\}.$$

3) Encryption: Encrypt(*MPK, ck_i, $\mathcal{T}$*)→ *(CT)* is run by DO for encryption. DO randomly selects κ symmetric keys $ck = \{ck_1, ck_2, \ldots, ck_\kappa\}$ to encrypt the corresponding κ message $E_{ck}(M) = \{E_{ck_1}(M_1), E_{ck_2}(M_2), \ldots, E_{ck_k}(M_\kappa)\}$, and hierarchically encrypts the symmetric keys ck. DO sets up a hierarchical access control tree $\mathcal{T}$ and associates each symmetric key ck_i $(i = 1, \ldots, \kappa)$ with a level node L_i in $\mathcal{T}$, where $s_i \in \mathbb{Z}_p$ is the secret numbers to the level nodes. Moreover, DO randomly chooses a data noise vector $\varepsilon = \{\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_\kappa\}$ ($\varepsilon_i \in \mathbb{Z}_p$ for each level node L_i). Then, it calculates level ciphertexts as

$$\left\{ \forall i \in L_i : C_{L_i} = ck_i \cdot e(g, g)^{\alpha(s_i+\varepsilon_i)} \right.$$
$$\left. C_{L_i}' = f_1^{(s_i+\varepsilon_i)}, C_{L_i}'' = f_2^{(s_i+\varepsilon_i)}, C_{L_i}''' = f_3^{\varepsilon_i} \right\}.$$

Then, DO chooses a $(k_x - 1)$-order polynomial function $q_x()$, and assigns values beginning from $\mathcal{T}$'s root node. DO sets $q_x(0) = s_i$ when x is a level node for ck_i. Otherwise, $q_x(0) = q_{\text{parent}(x)}(\textit{index}(x))$. We suppose that $\mathbb{X}$ is the leaf node set in $\mathcal{T}$. DO chooses the attribute $\text{att}(x) \in \mathbb{X}$, and calculates the attribute ciphertexts as

$$\{\overset{\frown}{C}_x = g^{q_x(0)}, \overset{\smile}{C}_x = H_1(\text{att}(x))^{q_x(0)}\}.$$

DO sets the transmission ciphertext of each transport node x for hierarchical decryption. For each child node $ch_{x,j} \in TN - CT(x) = \{ch_{x,1}, ch_{x,2}, \ldots\}$, it randomly chooses $l_{x,j} \in \mathbb{Z}_p$ and calculates the transmission ciphertexts with $v_x = H_2(e(g, g)^{\alpha(q_x(0)+\varepsilon_x)})$ as

$$\left\{ \bar{C}_{x,ch_{x,j}} = e\left(g, g^{v_x}\right)^{l_{x,j}} e(g, g)^{\alpha\left(q_{ch_{x,j}}(0)+\varepsilon_{ch_{x,j}}\right)} \right.$$
$$\left. \bar{C}_{x,ch_{x,j}}' = g^{l_{x,j}} \right\}.$$

$\{\mathcal{T}, C_{L_i}, C_{L_i}', C_{L_i}'', C_{L_i}''', \overset{\frown}{C}_x, \overset{\smile}{C}_x, \bar{C}_{x,ch_{x,j}}, \bar{C}_{x,ch_{x,j}}'\}$ is the ciphertext CT. According to Fig. 3, We give an example of the ciphertext associated with an access tree consisting of five files

$$CT = \left\{ \begin{array}{c} \mathcal{T}, C_{L_1}, C_{L_1}', C_{L_1}'', C_{L_1}''', C_{L_2}, C_{L_2}', C_{L_2}'', \\ C_{L_2}''', C_{L_3}, C_{L_3}', C_{L_3}'', C_{L_3}''', C_{L_4}, \\ C_{L_4}', C_{L_4}'', C_{L_4}''', C_{L_5}, C_{L_5}', C_{L_5}'', C_{L_5}''', \\ \overset{\frown}{C}_B, \overset{\smile}{C}_B, \overset{\frown}{C}_C, \overset{\smile}{C}_C, \overset{\frown}{C}_D, \overset{\smile}{C}_D, \overset{\frown}{C}_E, \overset{\smile}{C}_E, \\ \overset{\frown}{C}_F, \overset{\smile}{C}_F, \overset{\frown}{C}_G, \overset{\smile}{C}_G, \overset{\frown}{C}_H, \overset{\smile}{C}_H, \overset{\frown}{C}_I, \overset{\smile}{C}_I, \\ \bar{C}_{L_1,L_2}, \bar{C}_{L_1,L_2}', \bar{C}_{L_1,L_3}, \bar{C}_{L_1,L_3}', \bar{C}_{L_1,A}, \\ \bar{C}_{L_1,A}', \bar{C}_{L_2,L_4}, \bar{C}_{L_2,L_4}', \bar{C}_{A,L_5}, \bar{C}_{A,L_5}' \end{array} \right\}.$$

4) Decryption: Decrypt(*MPK, CT, SK_i*)→ (ck_i) is run by DU to decrypt the ciphertext. DU computes F_x = *DecryptNode(SK, CT, x)* for each leaf node x with $\text{att}_{i,j} = \text{att}(x)$

$$\begin{aligned} F_x = \textit{DecryptNode(SK, CT, x)} &= \frac{e\left(\overset{\frown}{D}_{i,j}, \overset{\frown}{C}_x\right)}{e\left(\overset{\smile}{D}_{i,j}, \overset{\smile}{C}_x\right)} \\ &= \frac{e\left(g^{r_i+\omega_1} H_1\left(\text{att}_{i,j}\right)^{r_{i,j}}, g^{q_x(0)}\right)}{e\left(g^{r_{i,j}}, H_1(\text{att}(x))^{q_x(0)}\right)} \\ &= e(g, g)^{(r_i+\omega_1)q_x(0)}. \end{aligned}$$

For nonleaf node x, DU calculates F_x with the Lagrange coefficient $\Delta_{i,S_x'}(0)$ and F_z. S_x is the arbitrary k_x child nodes z of x. If $S_x \subseteq S_i$ does not exist, $F_x = \perp$. Otherwise, DU can calculate F_x with $S_x' = \{i = \textit{index}(z) : z \in S_x\}$

$$\begin{aligned} F_x &= \prod_{z \in S_x} F_z^{\Delta_{i,S_x'}(0)} \\ &= \prod_{z \in S_x} \left(e(g, g)^{(r_i+\omega_1)q_z(0)}\right)^{\Delta_{i,S_x'}(0)} \\ &= \prod_{z \in S_x} \left(e(g, g)^{(r_i+\omega_1)q_x(i)}\right)^{\Delta_{i,S_x'}(0)} \\ &= e(g, g)^{(r_i+\omega_1)q_x(0)}. \end{aligned}$$

If DU's attribute set meets the part or whole hierarchical tree $\mathcal{T}$ at $\mathcal{T}_{L_i}$, DU can get the corresponding F_{L_i} of the Level nodes L_i. Then, DU calculates $\tilde{F}_i$

$$\begin{aligned} \tilde{F}_i &= \frac{e\left(C_{L_i}', D_i\right) \cdot e\left(C_{L_i}''', E_i'\right) \cdot F_{L_i}}{e\left(C_{L_i}'', E_i\right)} \\ &= \frac{e\left(f_1^{(s_i+\varepsilon_i)}, g^{\frac{\alpha+\theta}{\beta_1}}\right) \cdot e\left(f_3^{\varepsilon_i}, g^{\frac{r_i+\omega_1}{\beta_3}}\right) \cdot e(g, g)^{(r_i+\omega_1)s_i}}{e\left(f_2^{(s_i+\varepsilon_i)}, g^{\frac{\theta+r_i+\omega_1}{\beta_2}}\right)} \\ &= \frac{e(g, g)^{(\alpha+\theta)(s_i+\varepsilon_i)} \cdot e(g, g)^{(r_i+\omega_1)\varepsilon_i} \cdot e(g, g)^{(r_i+\omega_1)s_i}}{e(g, g)^{(\theta+r_i+\omega_1)(s_i+\varepsilon_i)}} \\ &= e(g, g)^{\alpha(s_i+\varepsilon_i)}. \end{aligned}$$

When DU can recover $\tilde{F}_i$ of the level node L_i, DU can conveniently compute $\tilde{F}_{ch_{L_i,j}}$ of the child node $ch_{L_i,j}$ $(j = 1, 2, \ldots)$ in subtree $\mathcal{T}_{L_i}$ through the transmission ciphertexts

$$\begin{aligned} \tilde{F}_{ch_{L_i,j}} &= \bar{C}_{L_i,ch_{L_i,j}} / e\left(\bar{C}_{L_i,ch_{L_i,j}}', g\right)^{H_2(\tilde{F}_i)} \\ &= e\left(g, g^{H_2\left(e(g,g)^{\alpha\left(q_{L_i}(0)+\varepsilon_{L_i}\right)}\right)}\right)^{l_{L_i,j}} \\ &\quad \cdot e(g, g)^{\alpha\left(q_{ch_{L_i,j}}(0)+\varepsilon_{ch_{L_i,j}}\right)} \\ &\quad \cdot e\left(g^{l_{L_i,j}}, g\right)^{-H_2\left(e(g,g)^{\alpha\left(q_{L_i}(0)+\varepsilon_{L_i}\right)}\right)} \\ &= e(g, g)^{\alpha\left(q_{ch_{L_i,j}}(0)+\varepsilon_{ch_{L_i,j}}\right)}. \end{aligned}$$

Setup*(λ) $\rightarrow$ (MPK, MSK)*. CA inputs λ and chooses $e : G_0 \times G_0 \rightarrow G_T$ with G_0, g and p. This algorithm defines H_1: $\{0,1\}^*$ $\rightarrow G_0$, $H_2 : G_T \rightarrow \mathbb{Z}_p$ and $H_3 : \{0,1\}^* \rightarrow \mathbb{Z}_p$. Then, CA randomly selects $\alpha, \beta_1, \beta_2, \beta_3 \in \mathbb{Z}_p$, and published *MPK* is:

$$\{g, G_0, e, H_1, H_2, f_1 = g^{\beta_1}, f_2 = g^{\beta_2}, f_3 = g^{\beta_3}, e(g,g)^{\alpha}\}$$

Let $h_1 = g^{\beta_1^{-1}}$, $h_2 = g^{\beta_2^{-1}}$ and $h_3 = g^{\beta_3^{-1}}$. CA ramdomly chooses $\theta \in \mathbb{Z}_p$ as the value of file hierarchy, and *MSK* is:

$$\{\alpha, \beta_1, \beta_2, \beta_3, \theta, H_3\}$$

Keygen*(MPK, MSK, S_i) $\rightarrow$ (SK_i)*. For the i-th user with user identification uid, CA randomly chooses $r_i, \omega_1 \in \mathbb{Z}_p$ where $\omega_1 = H_3(uid)$ for user, and $r_{i,j} \in \mathbb{Z}_p$ ($j \in [1, N_i]$) for each attribute in $S_i = \{att_{i,1}, \ldots, att_{i,N_i}\}$. CA computes the translation key $TK_{u_k}^{out}$ with randon number $zk \in \mathbb{Z}_p$ as:

$$\{D_i = g^{\frac{\alpha+\theta}{\beta_1 zk}}, E_i = g^{\frac{\theta+r_i+\omega_1}{\beta_2 zk}}, E_i^{'} = g^{\frac{r_i+\omega_1}{\beta_3 zk}}, \forall j \in [1, N_i] : \overset{\frown}{D}_{i,j} = g^{\frac{r_i+\omega_1}{zk}} H_1(att_{i,j})^{\frac{r_{i,j}}{zk}}, \overset{\smile}{D}_{i,j} = g^{\frac{r_{i,j}}{zk}}\}$$

Next, CA pass translation keys $TK_{u_k}^{out}$ to CS and pass private keys $SK_{u_k}^{out} = \left\{zk, TK_{u_k}^{out}\right\}$ to user u_k.

Encrypt*(MPK, ck_i, $\mathcal{T}$)$\rightarrow$ (CT)*. DO randomly selects κ symmetric keys $ck = \{ck_1, ck_2, \ldots, ck_\kappa\}$ to encrypt κ message $E_{ck}(M) = \{E_{ck_1}(M_1), E_{ck_2}(M_2), \ldots, E_{ck_k}(M_\kappa)\}$, and hierarchically encrypts the symmetric keys ck. DO sets up a hierarchical access control tree $\mathcal{T}$ and associates each symmetric key ck_i ($i = 1, \ldots, \kappa$) with a level node L_i in $\mathcal{T}$, where $s_i \in \mathbb{Z}_p$ is the secret numbers to the level nodes. Moreover, DO randomly chooses a data noise vector $\varepsilon = \{\varepsilon_i\}$ ($\varepsilon_i \in \mathbb{Z}_p$for each level node L_i). Then, it calculates level ciphertexts as:

$$\{\forall i \in L_i : C_{L_i} = ck_i \cdot e(g,g)^{\alpha(s_i+\varepsilon_i)}, C_{L_i}^{'} = f_1^{(s_i+\varepsilon_i)}, C_{L_i}^{''} = f_2^{(s_i+\varepsilon_i)}, C_{L_i}^{'''} = f_3^{\varepsilon_i}\}$$

Then, DO chooses a $(k_x - 1)$-order polynomial function $q_x()$ and sets $q_x(0) = s_i$. DO chooses $att(x) \in \mathbb{X}$ and calculates:

$$\{\overset{\frown}{C}_x = g^{q_x(0)}, \overset{\smile}{C}_x = H_1(att(x))^{q_x(0)}\}$$

DO sets the transmission ciphertext of each transport node x for hierarchical decryption. For each child node $ch_{x,j} \in$ $TN - CT(x) = \{ch_{x,1}, \ldots\}$, it randomly chooses $l_{x,j} \in \mathbb{Z}_p$ and calculates the transmission ciphertexts with $\nu_x =$ $H_2\left(e(g,g)^{\alpha(q_x(0)+\varepsilon_x)}\right)$ as:

$$\{\bar{C}_{x,ch_{x,j}} = e(g, g^{\nu_x})^{l_{x,j}} e(g,g)^{\alpha\left(q_{ch_{x,j}}(0)+\varepsilon_{ch_{x,j}}\right)}, \bar{C}_{x,ch_{x,j}}^{'} = g^{l_{x,j}}\}$$

$\left\{\mathcal{T}, C_{L_i}, C_{L_i}^{'}, C_{L_i}^{''}, C_{L_i}^{'''}, \overset{\frown}{C}_x, \overset{\smile}{C}_x, \bar{C}_{x,ch_{x,j}}, \bar{C}_{x,ch_{x,j}}^{'}\right\}$ is the ciphertext *CT*.

Decrypt*(MPK, CT, SK_i)$\rightarrow$ (ck_i)*. CS computes $F_x =$ *DecryptNode(TK, CT, x)* for each leaf node x with $TK_{u_k}^{out}$ and $att_{i,j}$ $= att(x)$.

$$F_x = \textit{DecryptNode(TK, CT, x)} = \frac{e\left(\overset{\frown}{D}_{i,j}, \overset{\frown}{C}_x\right)}{e\left(\overset{\smile}{D}_{i,j}, \overset{\smile}{C}_x\right)} = \frac{e\left(g^{\frac{r_i+\omega_1}{zk}} H_1(att_{i,j})^{\frac{r_{i,j}}{zk}}, g^{q_x(0)}\right)}{e\left(g^{\frac{r_{i,j}}{zk}}, H_1(att(x))^{q_x(0)}\right)} = e(g,g)^{\frac{(r_i+\omega_1)}{zk} q_x(0)}$$

For non-leaf node x, CS calculates F_x with the Lagrange coefficient $\Delta_{i,S_x^{'}}(0)$ and F_z. S_x is the arbitrary k_x child nodes z of x. If $S_x \subseteq S_i$ does not exist, $F_x = \perp$. Otherwise, CS can calculate F_x with $S_x^{'} = \{i = index(z) : z \in S_x\}$.

$$F_x = \prod_{z \in S_x} F_z^{\Delta_{i,S_x^{'}}(0)} = e(g,g)^{\frac{(r_i+\omega_1)}{zk} q_x(0)}$$

If DU's attribute set meets the part or whole hierarchical tree $\mathcal{T}$ at $\mathcal{T}_{L_i}$, CS can get the corresponding F_{L_i} of the Level nodes L_i. Then, CS calculates $CT^{out} = \left\{\forall i \in L_i : C_{L_i} = ck_i \cdot e(g,g)^{\alpha(s_i+\varepsilon_i)}, \tilde{F}_i\right\}$.

$$\tilde{F}_i = \frac{e\left(C_{L_i}^{'}, D_i\right) \cdot e\left(C_{L_i}^{'''}, E_i^{'}\right) \cdot F_{L_i}}{e\left(C_{L_i}^{''}, E_i\right)} = \frac{e\left(f_1^{(s_i+\varepsilon_i)}, g^{\frac{\alpha+\theta}{\beta_1 zk}}\right) \cdot e\left(f_3^{\varepsilon_i}, g^{\frac{r_i+\omega_1}{\beta_3 zk}}\right) \cdot e(g,g)^{\frac{(r_i+\omega_1)}{zk} s_i}}{e\left(f_2^{(s_i+\varepsilon_i)}, g^{\frac{\theta+r_i+\omega_1}{\beta_2 zk}}\right)} = e(g,g)^{\frac{\alpha(s_i+\varepsilon_i)}{zk}}$$

When CS can recover $\tilde{F}_i$ of the level node L_i, CS can pass ciphertexts CT^{out} to DU. Finally, DU recovers ck_i as follows.

$$\frac{C_{L_i}}{\left(\tilde{F}_i\right)^{zk}} = \frac{ck_i \cdot e(g,g)^{\alpha(s_i+\varepsilon_i)}}{\left(e(g,g)^{\frac{\alpha(s_i+\varepsilon_i)}{zk}}\right)^{zk}} = ck_i$$

Fig. 5. Improved CR-FH-CPABE scheme with outsourced decryption.

Finally, DU recovers ck_i as follows:

$$\frac{C_{L_i}}{\tilde{F}_i} = \frac{ck_i \cdot e(g,g)^{\alpha(s_i+\varepsilon_i)}}{e(g,g)^{\alpha(s_i+\varepsilon_i)}} = ck_i.$$

C. Construction of Improved CR-FH-CPABE Scheme

Based on Green et al. [39], we provide the details of the algorithms in our improved CR-FH-CPABE scheme with outsourced decryption as shown in Fig. 5. With the help of CS, DU can save a lot of computational costs. In addition, DU can modify the value of $zk_{\text{new}} = zk \cdot zk'$ ($zk' \in \mathbb{Z}_p$) to update the translation key $TK_{\text{new}} = (TK)^{(1/zk')}$ stored in CS.

VI. Security Proof

Based on the security of the original CP-ABE [12], we provide the security proofs of our CR-FH-CPABE scheme, which is CPA secure.

A. Security Model of CP-ABE

We give the security model between a challenger $\mathcal{C}$ and an adversary $\mathcal{A}$.

1) *Init:* $\mathcal{A}$ chooses an access structure $\mathbb{A}^*$ to challenge and sends it to $\mathcal{C}$.
2) *Setup:* Algorithm *Setup*() is run by $\mathcal{C}$, then $\mathcal{C}$ sends MPK to $\mathcal{A}$.

3) *QueryPhase 1:* $\mathcal{A}$ repeatedly queries $\mathcal{C}$ for private keys for the chosen attribute sets $S_1, S_2, \ldots, S_{q_1}$, where each S_i does not satisfy $\mathbb{A}^*$. The algorithm *Keygen*() is run by $\mathcal{C}$, then $\mathcal{C}$ sends the private keys back for responses.
4) *Challenge:* When $\mathcal{A}$ completes *QueryPhase 1*, it selects two equal-length messages msg_0 and msg_1 to challenge. $\mathcal{C}$ randomly chooses msg_μ with $\mu \in \{0, 1\}$ and uses algorithm *Encrypt*() to encrypt msg_μ under $\mathbb{A}^*$. $\mathcal{C}$ sends CT^* to $\mathcal{A}$.
5) *QueryPhase 2:* This phase is the same as the description of *QueryPhase 1* with the chosen attribute sets $S_{q_1+1}, S_{q_1+2}, \ldots, S_{q_2}$.
6) *Guess:* Finally, $\mathcal{A}$ returns a value $\mu' \in \{0, 1\}$ as the result, and $\mathcal{A}$ wins this game with advantage $|\Pr[\mu' = \mu] - (1/2)|$ when $\mu' = \mu$.

Definition 2: The CP-ABE scheme is secure against CPA if no probabilistic polynomial-time (PPT) adversary wins this game with nonnegligible advantage $|\Pr[\mu' = \mu] - (1/2)|$.

B. Security Proof for CR-FH-CPABE

Theorem 1: Assuming that there is no PPT adversary with a nonnegligible advantage in breaking the security of the CP-ABE scheme, then there is no PPT adversary that can break our CR-FH-CPABE scheme with a nonnegligible advantage.

Proof: Suppose there exists an adversary $\mathcal{A}$ with a nonnegligible advantage $Adv_{\mathcal{A}}$ in breaking our CR-FH-CPABE scheme; we construct an adversary $\mathcal{B}$ to break CP-ABE scheme by using $\mathcal{A}$. $\mathcal{B}$ acts as the challenger of $\mathcal{A}$, also called simulator $\mathcal{B}$.

1) *Init:* In the CP-ABE scheme, $\mathcal{B}$ receives MPK' without knowing MSK'

$$\mathrm{MPK}' = \left\{g, G_0, f = g^{\beta}, h = g^{\beta^{-1}}, e(g, g)^{\alpha}\right\}$$
$$\mathrm{MSK}' = \left\{\beta, g^{\alpha}\right\}.$$

2) *Setup:* For the CR-FH-CPABE scheme, $\mathcal{B}$ selects random values $x_1, x_2 \in \mathbb{Z}_p$ and computes MPK with MPK', where $\beta_1 = \beta$, $\beta_2 = x_1\beta$, and $\beta_3 = x_2\beta$. $\mathcal{B}$ sends MPK to $\mathcal{A}$

$$\mathrm{MPK} = \{g, G_0, f_1 = g^{\beta}, f_2 = g^{x_1\beta}, f_3 = g^{x_2\beta}, e(g, g)^{\alpha}\}.$$

3) *QueryPhase 1:* $\mathcal{B}$ responds to the queries for key generation of $\mathcal{A}$. Suppose $\mathcal{B}$ repeatedly receives the chosen sets $S_1, S_2, \ldots, S_{q_1}$, where each S_i does not satisfy $\mathbb{A}^*$. To answer the one query S_i $(i \in [1, q_1])$, $\mathcal{B}$ queries to $\mathcal{C}$ of the CP-ABE scheme for key generation. It makes the key queries for an attribute set twice and obtains two different SK_i^0 and SK_i^1 corresponding to S_i as

$$SK_i^0 = \Bigg\{D^0 = g^{\frac{\alpha + r_i}{\beta}}$$
$$\forall j \in [1, N_i] : D_j^0 = g^{r_i} H_1\big(\mathrm{att}_{i,j}\big)^{r_{i,j}}, D_{i,j}'^0 = g^{r_{i,j}}\Bigg\}$$
$$SK_i^1 = \Bigg\{D^1 = g^{\frac{\alpha + r_i'}{\beta}}$$
$$\forall j \in [1, N_i] : D_j^1 = g^{r_i'} H_1\big(\mathrm{att}_{i,j}\big)^{r_{i,j}'}, D_{i,j}'^1 = g^{r_{i,j}'}\Bigg\}$$

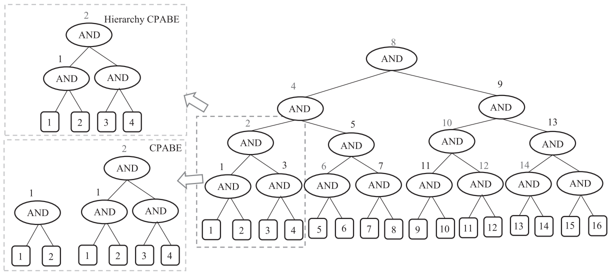


Fig. 6. Example tree in simulation with CR-FH-CPABE.

where the attribute $\mathrm{att}_{i,j} \in S_i$ and random numbers $r_i, r_i', r_{i,j}, r_{i,j}' \in \mathbb{Z}_p$. From SK_i^0 and SK_i^1, $\mathcal{B}$ can calculate $D^0/D^1 = g^{(r_i - r_i'/\beta)}$. It selects $t_i, t_{i,j} \in \mathbb{Z}_p$ and sets $\theta = r_i$ and $r_i + \omega_1 = t_i - r_i'$. Then, $\mathcal{B}$ can derive SK_i^* as

$$SK_i^* = \Bigg\{D_i = g^{\frac{\alpha + r_i}{\beta}}, E_i = g^{\frac{r_i - r_i' + t_i}{x_1\beta}}$$
$$E_i' = g^{t_i - r_i'} \cdot h^{t_{i,j} - r_{i,j}'}, \left(E_i'' = g^{t_{i,j} - r_{i,j}'}\right)$$
$$\forall j \in [1, N_i] : \widehat{D}_{i,j} = g^{t_i - r_i'} H_1\big(\mathrm{att}_{i,j}\big)^{t_{i,j} - r_{i,j}'}$$
$$\breve{D}_{i,j} = g^{t_{i,j} - r_{i,j}'}\Bigg\}.$$

Note that E_i' has no parameter β and it is one value of $\widehat{D}_{i,j}$ with E_i''. $\mathcal{B}$ will set the corresponding solutions in ciphertext to train $\mathcal{A}$. Then, $\mathcal{B}$ sends the private keys SK_i^* corresponding to each S_i to $\mathcal{A}$.

4) *Challenge:* When $\mathcal{A}$ completes *QueryPhase 1*, it selects two equal-length messages msg_0 and msg_1 to challenge. $\mathcal{B}$ sends them to $\mathcal{C}$ of CP-ABE scheme, then it gets the challenge CT

$$CT = \Bigg\{\mathcal{T}, C_0 = msg_u \cdot e(g, g)^{\alpha s}, C_0' = f^s$$
$$\forall x \in \mathbb{X}: C_x = g^{q_x(0)}, C_x' = H_1(\mathrm{att}(x))^{q_x(0)}\Bigg\}.$$

Then, $\mathcal{B}$ chooses $\varepsilon \in \mathbb{Z}_p$ and calculates ciphertext CT^* for $\mathcal{A}$ to challenge. $\mathcal{B}$ uses CT to construct a hierarchical access control structure with only one level node, so there is no need to include transmission ciphertext in CT^*

$$CT^* = \Bigg\{\mathcal{T}, C_{L_1} = msg_u \cdot e(g, g)^{\alpha(s+\varepsilon)}$$
$$C_{L_1}' = f_1^{(s+\varepsilon)}, C_{L_1}'' = f_2^{(s+\varepsilon)}, C_{L_1}''' = f_3^{\varepsilon}$$
$$\forall x \in \mathbb{X}: \widehat{C}_x = g^{q_x(0)}, \breve{C}_x = H_1(\mathrm{att}(x))^{q_x(0)}\Bigg\}.$$

Note that E_i' is fixed value in SK_i^* of our CR-FH-CPABE scheme, and $\mathcal{B}$ can set CT' with ε and the public hash value $h = H_1(\mathrm{att})$ in the previous training

$$CT' = \Bigg\{\mathcal{T}, C_{L_1} = msg_u \cdot e(g, g)^{\alpha(s+\varepsilon)}$$
$$C_{L_1}' = f_1^{(s+\varepsilon)}, C_{L_1}'' = f_2^{(s+\varepsilon)}, C_{L_1}''' = f_3^{\varepsilon}$$

TABLE III
COMPARISONS OF FUNCTION REALIZATION

Scheme	Multi-authority	Transmission ciphertext	CPA secure	Collusion resistance	Outsourced decryption
Wang el al. [9]	✗	✓	✗	✗	✗
Li el al. [34]	✗	✓	✓	✗	✗
Xiao el al. [35]	✗	✗	✓	✗	✗
Zaghloul el al. [36]	✗	✗	✓	✗	✗
Xu el al. [38]	✓	✗	✓	✗	✗
Fu el al. [37]	✗	✗	✓	✗	✗
Jiang el al. [3]	✗	✓	✗	✗	✗
Guo el al. [33]	✓	✓	✗	✗	✗
Liu el al. [4]	✓	✓	✗	✗	✗
CR-FH-CPABE	✗	✓	✓	✓	✗
Improved CR-FH-CPABE	✗	✓	✓	✓	✓

TABLE IV
COMPUTATION OVERHEAD

Scheme	Encryption	Decryption (with transmission ciphertext)
CP-ABE [12]	$(2\lvert\mathcal{T}_1\rvert + 2\lvert\mathcal{T}_2\rvert + \cdots + 2\lvert\mathcal{T}_k\rvert + k)\, E_{G_0} + kE_{G_T}$	$(2k\lvert\mathcal{T}_1\rvert + 2k\lvert\mathcal{T}_2\rvert + \cdots + 2k\lvert\mathcal{T}_k\rvert + k)\, P + (\lvert\mathcal{T}_1\rvert + \lvert\mathcal{T}_2\rvert + \cdots + \lvert\mathcal{T}_k\rvert)\, E_{G_T}$
FH-CP-ABE [9]	$(2\lvert\mathcal{T}_{ln}\rvert + k)\, E_{G_0} + (j\lvert\mathcal{T}_{tn}\rvert + k)\, E_{G_T}$	$(2\lvert\mathcal{T}_{ln}\rvert + k)\, P + \lvert\mathcal{T}_{ln}\rvert E_{G_T}$ or $(2\lvert\mathcal{T}_{ln}\rvert + 1)\, P + \lvert\mathcal{T}_{ln}\rvert E_{G_T}$
EFH-CP-ABE [34]	$(2\lvert\mathcal{T}_{ln}\rvert + 2j\lvert\mathcal{T}_{tn}\rvert + k)\, E_{G_0} + (j\lvert\mathcal{T}_{tn}\rvert + k)\, E_{G_T} + j\lvert\mathcal{T}_{tn}\rvert P$	$(2\lvert\mathcal{T}_{ln}\rvert + k)\, P + \lvert\mathcal{T}_{ln}\rvert E_{G_T}$ or $(2\lvert\mathcal{T}_{ln}\rvert + j\lvert\mathcal{T}_{tn}\rvert + 1)\, P + (j\lvert\mathcal{T}_{tn}\rvert + \lvert\mathcal{T}_{ln}\rvert)\, E_{G_T}$
CR-FH-CPABE	$(2\lvert\mathcal{T}_{ln}\rvert + 2j\lvert\mathcal{T}_{tn}\rvert + 3k)\, E_{G_0} + (j\lvert\mathcal{T}_{tn}\rvert + k)\, E_{G_T} + j\lvert\mathcal{T}_{tn}\rvert P$	$(2\lvert\mathcal{T}_{ln}\rvert + 3k)\, P + \lvert\mathcal{T}_{ln}\rvert E_{G_T}$ or $(2\lvert\mathcal{T}_{ln}\rvert + j\lvert\mathcal{T}_{tn}\rvert + 3)\, P + (j\lvert\mathcal{T}_{tn}\rvert + \lvert\mathcal{T}_{ln}\rvert)\, E_{G_T}$
Improved CR-FH-CPABE	$(2\lvert\mathcal{T}_{ln}\rvert + 2j\lvert\mathcal{T}_{tn}\rvert + 3k)\, E_{G_0} + (j\lvert\mathcal{T}_{tn}\rvert + k)\, E_{G_T} + j\lvert\mathcal{T}_{tn}\rvert P$	kE_{G_T}

[*] k is the number of files. j is the maximum value of the size of *TN-CT*. E_{G_0} and E_{G_T} respectively represent the exponentiation time in G_0 and G_T. P indicates the time of bilinear pairing e. $\lvert\mathcal{T}\rvert$ is the attribute number of access tree $\mathcal{T}$, and $\lvert\mathcal{T}_{ln}\rvert$ and $\lvert T_{tn}\rvert$ are the value of leaf nodes and transport nodes of $\mathcal{T}$. $\mathcal{T}_i$ $(i = 1, \ldots, k)$ is the subtree of $\mathcal{T}$ at the root of level node.

$$\left.\begin{aligned} &C_{L_1}^{0''\prime} = g^{\varepsilon}, C_{L_1}^{1''\prime} = h^{\varepsilon} \\ &\forall x \in \mathbb{X} : \widehat{C}_x = g^{q_x(0)}, \breve{C}_x = H_1(\mathrm{att}(x))^{q_x(0)} \end{aligned}\right\}.$$

Then, $\mathcal{A}$ can calculate $e(g, g)^{(t_i - r'_i)\varepsilon}$ as follows:

$$\frac{e\left(C_{L_1}^{0''\prime}, E'_i\right)}{e\left(C_{L_1}^{1''\prime}, E''_i\right)} = \frac{e\left(g^{\varepsilon}, g^{t_i - r'_i} \cdot h^{t_{i,j} - r'_{i,j}}\right)}{e\left(h^{\varepsilon}, g^{t_{i,j} - r'_{i,j}}\right)} = e(g, g)^{(t_i - r'_i)\varepsilon}.$$

In this way, $\mathcal{A}$ can obtain the corresponding relationship between f_3^{ε} and $e(g, g)^{(t_i - r'_i)\varepsilon}$. Finally, $\mathcal{B}$ returns CT^* to $\mathcal{A}$.

5) *QueryPhase 2:* This phase is the same as the description of *QueryPhase 1* with the chosen attribute sets $S_{q_1+1}, S_{q_1+2}, \ldots, S_{q_2}$, and each S_i dose not satisfy the access structure $\mathbb{A}^*$.
6) *Guess:* Finally, $\mathcal{A}$ returns a value $\mu' \in \{0, 1\}$ as the result, then $\mathcal{B}$ concludes this game with the output μ'. According to the security model, $\mathcal{B}$ wins the game with advantage $Adv_{\mathcal{B}}(\lambda) = Adv_{\mathcal{A}}(\lambda) = |\Pr[\mu' = \mu] - (1/2)|$, which means the advantage of $\mathcal{B}$ against the CP-ABE scheme is nonnegligible, which completes the proof of Theorem 1. ∎

VII. Performance Evaluation

In this section, we conduct simulations of two schemes using Java language to demonstrate our practicality. We installed JPBC 2.0, JDK 19, and IntelliJ IDEA on a laptop with an Intel Core i7-10750h CPU and 16-GB RAM. We use the curve $y^2 = x^3 + x$ for pairing, and the values of rBits and qBits are 160 and 512 bits, respectively. These results were derived by carefully selecting and averaging over 100 experiments.

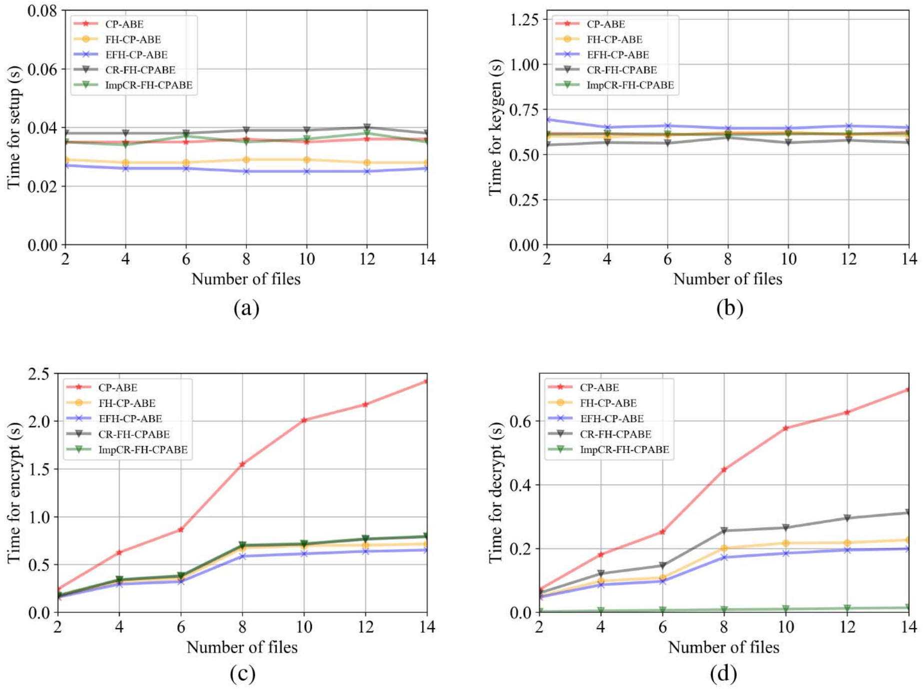
As shown in Table III, we have compared our CR-FH-CPABE scheme and improved CR-FH-CPABE scheme with the schemes [3], [4], [9], [33], [34], [35], [36], [37], [38] mentioned above. These schemes have different limitations in achieving hierarchical CPA security and preventing collusion attacks by malicious inside users. Schemes [35], [36], [37], and [38] do not consider transmission ciphertexts. From the perspective of leaked information through transmission ciphertexts, they assume that the CPA security proof holds in scenarios involving a single malicious user within the scheme. However, they overlook the possibility of user collusion attacks mentioned in this article. Furthermore, if a system already has security vulnerabilities due to user collusion attacks, introducing transmission ciphertexts can escalate the risks, similar to schemes [3], [4], [9], [33], and [34].

Our CR-FH-CPABE scheme can achieve secure hierarchical access control against collusion attacks while using the transmission ciphertext to save computation and storage overhead. Our improved CR-FH-CPABE scheme can provide outsourced decryption. We selected the representative FH-CP-ABE scheme [9], the original CP-ABE scheme [12], and EFH-CP-ABE scheme [34] for simulation.

A. Comparisons of Computation Overhead

We compare the computation overhead for users with the three schemes in Table IV. In the decryption part, we give two expressions: the former is to decrypt k files individually. The latter is to recover the highest-level node first, then decrypt other files with the transmission ciphertext. The original CP-ABE is computationally expensive for multiple encryptions. The latter three schemes use hierarchical encryption algorithms to improve computation efficiency. Compared with the FH-CP-ABE scheme, the EFH-CP-ABE scheme has made improvements and proposed a more secure transmission ciphertext. So, the encryption and decryption overhead will increase slightly. Our CR-FH-CPABE scheme solves the problem of user collusion, thereby increasing some computation overhead. However, its computation overhead is still similar to that of EFH-CP-ABE. Our improved CR-CPABE scheme outsources a large amount of computation to CS, so the decryption cost for users is very low.

We have implemented simulations for five different schemes and simplified the simulation methods. We conduct separate experiments by carefully controlling the number of files and



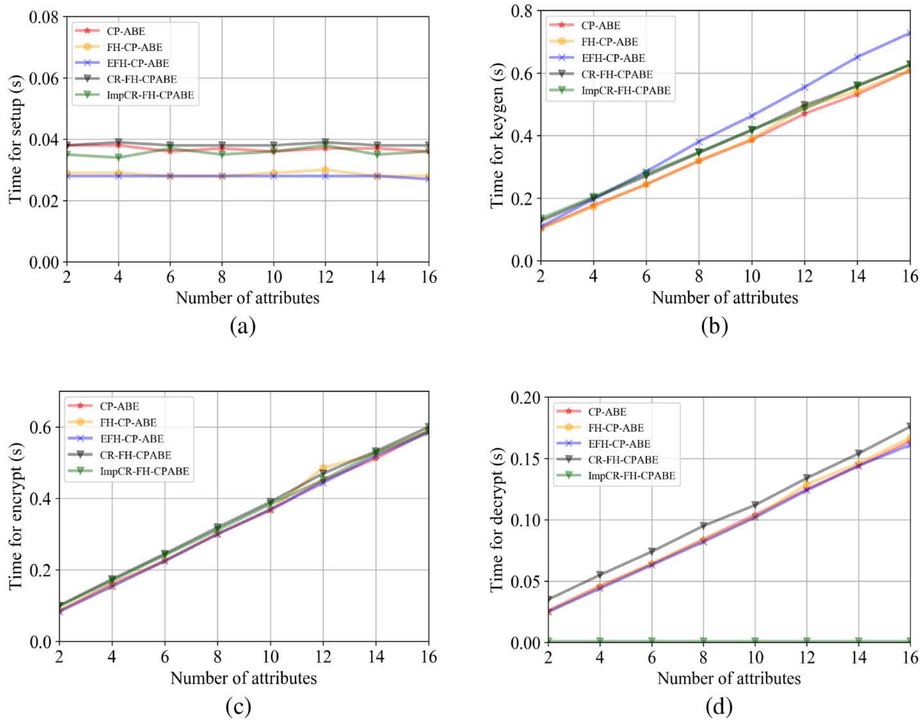
(a)

(b)

(c)

(d)

Fig. 7. Computation overhead with fixed attributes for CR-FH-CPABE. (a) Setup. (b) Keygen. (c) Encrypt. (d) Decrypt.



(a)

(b)

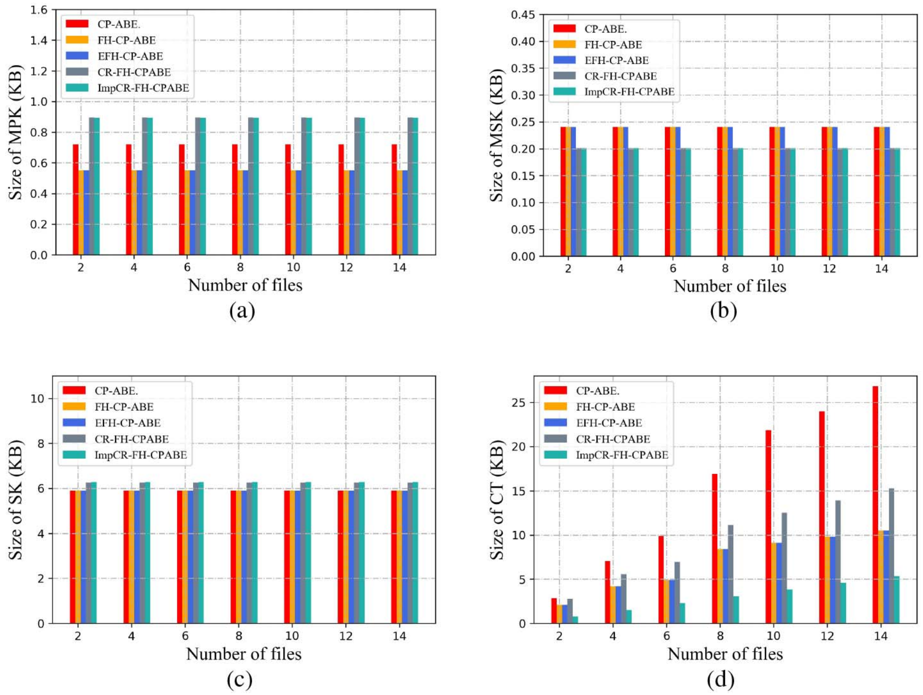
(c)

(d)

Fig. 8. Computation overhead with fixed files for CR-FH-CPABE. (a) Setup. (b) Keygen. (c) Encrypt. (d) Decrypt.

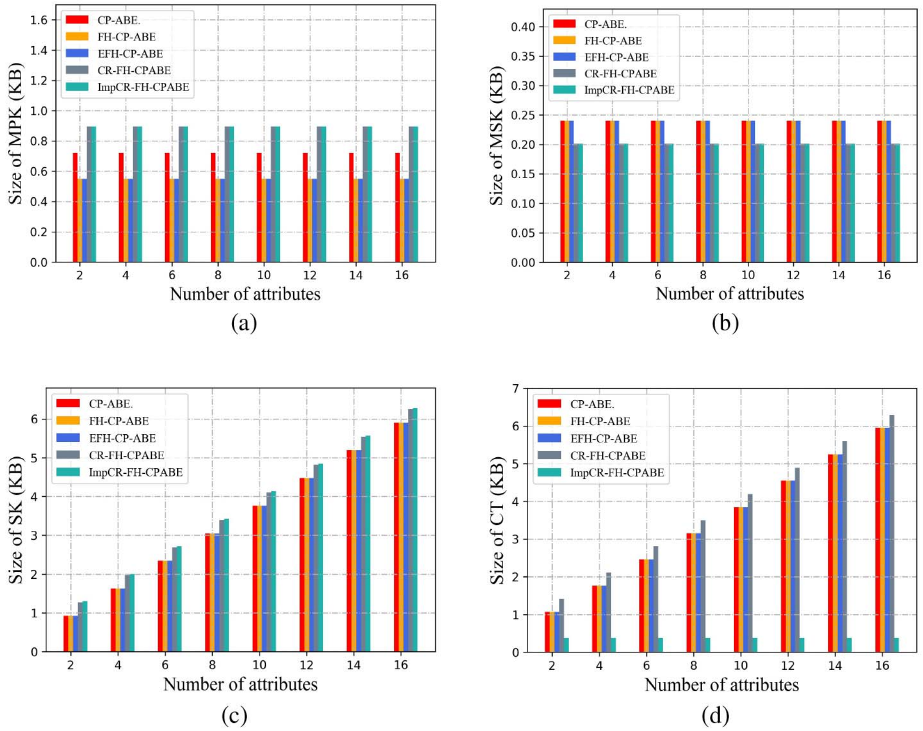
attributes. This allows us to observe and analyze both factors' impact on computation overhead. To consider the worst case in simulation, we choose the access control structure with only the *AND* gate.

First, we fix attribute number to 16, and its access control tree formed by these attributes can have a maximum of 15 level nodes. Therefore, we have chosen to increase the number of files from 2 to 14 for simulation, as shown in Fig. 6. Taking an example with two attributes, hierarchy CP-ABE only requires constructing one access structure and encrypting two files simultaneously. In contrast, CP-ABE requires creating two access structures and encrypting them separately. We sequentially increase file number for simulation as shown in Fig. 6. Their simulation results are shown in



(a) (b) (c) (d)

Fig. 9. Communication overhead with fixed attributes for CR-FH-CPABE. (a) MPK. (b) MSK. (c) SK. (d) CT.



(a) (b) (c) (d)

Fig. 10. Communication overhead with fixed files for CR-FH-CPABE. (a) MPK. (b) MSK. (c) SK. (d) CT.

Fig. 7(a)–(d). It shows that the computation overhead of hierarchical encryption does not change much when the number of files continues to grow. It can be seen that compared with the original CP-ABE scheme, hierarchical schemes have more significant advantages for access control with a large number of files. While ensuring secure hierarchical access, the result of our CR-FH-CPABE is similar to the other two hierarchical schemes. In decryption, our improved CR-FH-CPABE scheme has a powerful advantage, which is that the decryption time is almost constant.

Furthermore, we fix the file number, then calculation time is primarily influenced by the algorithm design of schemes. So we only select one file and vary the number of attributes from 2 to 16 for simulation. In this way, the access tree can be set as att_1 *AND* att_2 *AND*… *AND* $\text{att}_{|\mathcal{T}|}$. As shown in Fig. 8(a)–(d), it shows that the computation time of each

TABLE V
COMMUNICATION OVERHEAD

Scheme	*MPK*	*MSK*	*SK*	*CT*
CP-ABE [12]	$3L_{G_0} + L_{G_T}$	$L_{G_0} + L_{\mathbb{Z}_p}$	$(2\|S\| + 1)\, L_{G_0}$	$(2\|\mathcal{T}_1\| + 2\|\mathcal{T}_2\| + \cdots + 2\|\mathcal{T}_k\| + k)\, L_{G_0} + kL_{G_T}$
FH-CP-ABE [9]	$2L_{G_0} + L_{G_T}$	$L_{G_0} + L_{\mathbb{Z}_p}$	$(2\|S\| + 1)\, L_{G_0}$	$(2\|\mathcal{T}_{ln}\| + k)\, L_{G_0} + (j\|\mathcal{T}_{tn}\| + k)\, L_{G_T}$
EFH-CP-ABE [34]	$2L_{G_0} + L_{G_T}$	$L_{G_0} + L_{\mathbb{Z}_p}$	$(2\|S\| + 1)\, L_{G_0}$	$(2\|\mathcal{T}_{ln}\| + j\|\mathcal{T}_{tn}\| + k)\, L_{G_0} + (j\|\mathcal{T}_{tn}\| + k)\, L_{G_T}$
CR-FH-CPABE	$4L_{G_0} + L_{G_T}$	$5L_{\mathbb{Z}_p}$	$(2\|S\| + 3)\, L_{G_0}$	$(2\|\mathcal{T}_{ln}\| + j\|\mathcal{T}_{tn}\| + 3k)\, L_{G_0} + (j\|\mathcal{T}_{tn}\| + k)\, L_{G_T}$
Improved CR-FH-CPABE	$4L_{G_0} + L_{G_T}$	$5L_{\mathbb{Z}_p}$	$(2\|S\| + 3)\, L_{G_0} + L_{\mathbb{Z}_p}$	$2kL_{G_T}$

[*] k is the number of files. j is the maximum value of the size of *TN-CT*. L_{G_0}, L_{G_T}, $L_{\mathbb{Z}_p}$ respectively represent the size of G_0, G_T and $\mathbb{Z}_p$. $|S|$ represents the number of attributes for DU. $|\mathcal{T}_{ln}|$ and $|\mathcal{T}_{tn}|$ denote the number of leaf nodes and transport nodes in $\mathcal{T}$. $\mathcal{T}_i$ $(i = 1, \ldots, k)$ denotes the subtree of $\mathcal{T}$ rooted at level node.

increases as attribute number increases. In addition, our CR-FH-CPABE scheme has little difference in single computation overhead compared to the other three schemes, which is same with previous analysis. The decryption time of the improved CR-FH-CPABE scheme is stable constant.

B. Comparisons of Communication Overhead

As shown in Table V, we have also conducted the analysis of their communication overhead, and the simulation results are presented in Figs. 9 and 10(a)–(d). It is evident that the lengths of MPK, *MSK*, and *SK* are similar across five schemes. The hierarchical access control schemes exhibit minor communication overhead in terms of *CT*, and our improved CR-FH-CPABE scheme has the minimum ciphertext length because of outsourced decryption.

VIII. CONCLUSION

In this article, we proposed our secure CR-FH-CPABE scheme. First, we introduced the CR-FH-CPABE scheme, adding a data noise vector to resist user collusion attacks effectively. Building upon the hierarchical access, we introduced an improved CR-FH-CPABE scheme, which achieved outsourced decryption. Moreover, We found significant advantages of our schemes through security proofs and simulation experiments. We demonstrated our scheme to be CPA secure and showed outstanding performance through simulation results. In the future, we want to research user collaboration further to address the rapid cooperation in multiuser file hierarchy access control.

REFERENCES

[1] M. Armbrust et al., "A view of cloud computing," *Commun. ACM*, vol. 53, no. 4, pp. 50–58, 2010.

[2] J. Shen, H. Yang, P. Vijayakumar, and N. Kumar, "A privacy-preserving and untraceable group data sharing scheme in cloud computing," *IEEE Trans. Dependable Secure Comput.*, vol. 19, no. 4, pp. 2198–2210, Jul./Aug. 2022.

[3] S. Jiang, W. Guo, and G. Fan, "Hierarchy attribute-based encryption scheme to support direct revocation in cloud storage," in *Proc. IEEE/ACIS 16th Int. Conf. Comput. Inf. Sci. (ICIS)*, 2017, pp. 869–874.

[4] X. Liu, X. Yang, Y. Luo, L. Wang, and Q. Zhang, "Anonymous electronic health record sharing scheme based on decentralized hierarchical attribute-based encryption in cloud environment," *IEEE Access*, vol. 8, pp. 200180–200193, 2020.

[5] J. Zhang et al., "Efficient hierarchical data access control for resource-limited users in cloud-based e-Health," in *Proc. Int. Conf. Netw. Appl.*, 2019, pp. 319–324.

[6] K. Xue, N. Gai, J. Hong, D. S. L. Wei, P. Hong, and N. Yu, "Efficient and secure attribute-based access control with identical sub-policies frequently used in cloud storage," *IEEE Trans. Dependable Secure Comput.*, vol. 19, no. 1, pp. 635–646, Jan./Feb. 2022.

[7] Y. Zhang, R. H. Deng, S. Xu, J. Sun, Q. Li, and D. Zheng, "Attribute-based encryption for cloud computing access control: A survey," *ACM Comput. Survey*, vol. 53, no. 4, p. 83, 2020.

[8] A. Sahai and B. Waters, "Fuzzy identity based encryption," in *Proc. EUROCRYPT*, 2005, pp. 457–473.

[9] S. Wang, J. Zhou, J. K. Liu, J. Yu, J. Chen, and W. Xie, "An efficient file hierarchy attribute-based encryption scheme in cloud computing," *IEEE Trans. Inf. Forensics Security*, vol. 11, pp. 1265–1277, 2016.

[10] N. Chen, J. Li, Y. Zhang, and Y. Guo, "Efficient CP-ABE scheme with shared decryption in cloud storage," *IEEE Trans. Comput.*, vol. 71, no. 1, pp. 175–184, Jan. 2022.

[11] M. Rasori, M. L. Manna, P. Perazzo, and G. Dini, "A survey on attribute-based encryption schemes suitable for the Internet of Things," *IEEE Internet Things J.*, vol. 9, no. 11, pp. 8269–8290, Jun. 2022.

[12] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in *Proc. IEEE Symp. Security Privacy*, 2007, pp. 321–334.

[13] B. Waters, "Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization," in *Proc. 14th Int. Conf. Pract. Theory Public Key Cryptogr. Conf. Public Key Cryptogr.*, 2011, pp. 53–70.

[14] Y. Rouselakis and B. Waters, "Efficient statically-secure large-universe multi-authority attribute-based encryption," in *Proc. Int. Conf. Financ. Cryptogr. Data Secur.*, 2015, pp. 315–332.

[15] J. Li et al., "Multiauthority attribute-based encryption for assuring data deletion," *IEEE Syst. J.*, vol. 17, no. 2, pp. 2029–2038, Jun. 2023.

[16] A. B. Lewko and B. Waters, "Decentralizing attribute-based encryption," in *Proc. EUROCRYPT*, 2011, pp. 568–588.

[17] L. Zhang, T. Zhang, Q. Wu, Y. Mu, and F. Rezaeibagha, "Secure decentralized attribute-based sharing of personal health records with blockchain," *IEEE Internet Things J.*, vol. 9, no. 14, pp. 12482–12496, Jul. 2022.

[18] C. Ge, W. Susilo, J. Baek, Z. Liu, J. Xia, and L. Fang, "Revocable attribute-based encryption with data integrity in clouds," *IEEE Trans. Dependable Secure Comput.*, vol. 19, no. 5, pp. 2864–2872, Sep./Oct. 2022.

[19] H. Xiong, X. Huang, M. Yang, L. Wang, and S. Yu, "Unbounded and efficient revocable attribute-based encryption with adaptive security for cloud-assisted Internet of Things," *IEEE Internet Things J.*, vol. 9, no. 4, pp. 3097–3111, Feb. 2022.

[20] S. Chen, J. Li, Y. Zhang, and J. Han, "Efficient revocable attribute-based encryption with verifiable data integrity," *IEEE Internet Things J.*, early access, Oct. 23, 2023, doi: 10.1109/JIOT.2023.3325996.

[21] J. Li, W. Yao, J. Han, Y. Zhang, and J. Shen, "User collusion avoidance CP-ABE with efficient attribute revocation for cloud storage," *IEEE Syst. J.*, vol. 12, no. 2, pp. 1767–1777, Jun. 2018.

[22] C. Feng, K. Yu, M. Aloqaily, M. Alazab, Z. Lv, and S. Mumtaz, "Attribute-based encryption with parallel outsourced decryption for edge intelligent IoV," *IEEE Trans. Veh. Technol.*, vol. 69, no. 11, pp. 13784–13795, Nov. 2020.

[23] J. Li, Y. Wang, Y. Zhang, and J. Han, "Full verifiability for outsourced decryption in attribute based encryption," *IEEE Trans. Services Comput.*, vol. 13, no. 3, pp. 478–487, May/Jun. 2020.

[24] W. Zhang, Z. Zhang, H. Xiong, and Z. Qin, "PHAS-HEKR-CP-ABE: Partially policy-hidden CP-ABE with highly efficient key revocation in cloud data sharing system," *J. Ambient Intell. Humaniz. Comput.*, vol. 13, no. 1, pp. 613–627, 2022.

[25] Z. Zhang, J. Zhang, Y. Yuan, and Z. Li, "An expressive fully policy-hidden ciphertext policy attribute-based encryption scheme with credible verification based on blockchain," *IEEE Internet Things J.*, vol. 9, no. 11, pp. 8681–8692, Jun. 2022.

[26] D. Han, N. Pan, and K.-C. Li, "A traceable and revocable ciphertext-policy attribute-based encryption scheme based on privacy protection," *IEEE Trans. Dependable Secure Comput.*, vol. 19, no. 1, pp. 316–327, Jan./Feb. 2022.

[27] S. Xu et al., "Efficient ciphertext-policy attribute-based encryption with blackbox traceability," *Inf. Sci.*, vol. 538, pp. 19–38, Oct. 2020.

[28] C. Gentry and A. Silverberg, "Hierarchical ID-based cryptography," in *Proc. Int. Conf. Theory Appl. Cryptol. Inf. Security*, 2002, pp. 548–566.

[29] J. Li, Q. Wang, C. Wang, and K. Ren, "Enhancing attribute-based encryption with attribute hierarchy," *Mobile Netw. Appl.*, vol. 16, no. 5, pp. 553–561, 2011.

[30] H. Deng et al., "Ciphertext-policy hierarchical attribute-based encryption with short ciphertexts," *Inf. Sci.*, vol. 275, pp. 370–384, Aug. 2014.

[31] G. Wang, Q. Liu, and J. Wu, "Hierarchical attribute-based encryption for fine-grained access control in cloud storage services," in *Proc. ACM Conf. Comput. Commun. Security*, Oct. 2010, pp. 735–737.

[32] Z. Wan, J. Liu, and R. H. Deng, "HASBE: A hierarchical attribute-based solution for flexible and scalable access control in cloud computing," *IEEE Trans. Inf. Forensics Security*, vol. 7, pp. 743–754, 2012.

[33] R. Guo, X. Li, D. Zheng, and Y. Zhang, "An attribute-based encryption scheme with multiple authorities on hierarchical personal health record in cloud," *J. Supercomput.*, vol. 76, no. 7, pp. 4884–4903, 2020.

[34] J. Li, N. Chen, and Y. Zhang, "Extended file hierarchy access control scheme with attribute-based encryption in cloud computing," *IEEE Trans. Emerg. Topics Comput.*, vol. 9, no. 2, pp. 983–993, Apr.-Jun. 2021.

[35] M. Xiao, H. Li, Q. Huang, S. Yu, and W. Susilo, "Attribute-based hierarchical access control with extendable policy," *IEEE Trans. Inf. Forensics Security*, vol. 17, pp. 1868–1883, 2022.

[36] E. Zaghloul, K. Zhou, and J. Ren, "P-MOD: Secure privilege-based multilevel Organizational data-sharing in cloud computing," *IEEE Trans. Big Data*, vol. 6, no. 4, pp. 804–815, Dec. 2020.

[37] J. Fu and N. Wang, "A practical attribute-based document collection hierarchical encryption scheme in cloud computing," *IEEE Access*, vol. 7, pp. 36218–36232, 2019.

[38] Y. Xu, X. Dong, and J. Shen, "Multi-authority attribute-based encryption supporting hierarchal access policy and range policy," in *Proc. Int. Conf. Comput. Commun. Netw. Security (CCNS)*, 2020, pp. 81–86.

[39] M. Green, S. Hohenberger, and B. Waters, "Outsourcing the decryption of ABE ciphertexts," in *Proc. 20th USENIX Conf. Security*, 2011, pp. 1–6.

Yuhan Bai received the B.S. degree in electronic engineering from Xidian University, Xi'an, China, in 2018, where she is currently pursuing the Ph.D. degree in cyber security with the State Key Laboratory of Integrated Service Networks.

Her research interests are cloud security, IoT security, and network and information security.

Kai Fan (Member, IEEE) received the B.S. degree in telecommunication engineering, the M.S. degree in cryptography, and the Ph.D. degree in telecommunication and information system from Xidian University, Xi'an, China, in 2002, 2005, and 2007, respectively.

He is working as a Professor with the State Key Laboratory of Integrated Service Networks, Xidian University. He published over 70 papers in journals and conferences. He received nine Chinese patents. He has managed five national research projects. His research interests include IoT security and information security.

Kuan Zhang (Member, IEEE) received the B.S. degree in communication engineering and the M.S. degree in computer applied technology from Northeastern University, Shenyang, China, in 2009 and 2011, respectively, and the Ph.D. degree in electrical and computer engineering from the University of Waterloo, Waterloo, ON, Canada, in 2016.

He is working as an Assistant Professor with the Department of Electrical and Computer Engineering, University of Nebraska–Lincoln, Lincoln, NE, USA. He was a Postdoctoral Fellow with the University of Waterloo from 2016 to 2017. He has published over 50 papers in journals and conferences. His research interests include cyber security, big data, and cloud/edge computing.

Dr. Zhang was the recipient of the Best Paper Award at IEEE WCNC 2013 and Securecomm 2016.

Hui Li (Senior Member, IEEE) was born in Shaanxi, China, in 1968. He received the B.S. degree in radio electronics from Fudan University, Shanghai, China, in 1990, and the M.S. and Ph.D. degrees in telecommunications and information system from Xidian University, Xi'an, China, in 1993 and 1998, respectively.

He is currently a Professor with Xidian University. His research interests include network and information security.

Yintang Yang (Senior Member, IEEE) received the B.S. and M.S. degrees in semiconductor devices and microelectronics from Xidian University, Xi'an, China, in 1982 and 1984, respectively, and the Ph.D. degree in microelectronics and solid-state electronics from Xi'an Jiaotong University, Xi'an, in 2000.

Since 1984, he has been with the Department of Technical Physics, Xidian University, where he has been a Professor of Microelectronics and the Director of the Institute of Microelectronics since 1997. He has published three books and more than 140 journal and conference papers. His research interests include design methodologies and techniques for analog and mixed-signal CMOS integrated circuits, advanced semiconductor materials and devices, and micromechanical system and sensors.

Prof. Yang received the National Science Fund for Distinguished Young Scholars from the National Natural Science Foundation of China in 2007. He has been on the editorial board for four journals.