An Efficient File Hierarchy based E-health system with Clinical Document Architecture in Cloud Computing

R.Janarthanan, Associate Professor,CSE, T.J.S. Engineering college
S.Prathiba, Student, CSE, T.J.S. Engineering college
S.Preethi, Student, CSE, T.J.S. Engineering college
P.Priya, Student, CSE, T.J.S. Engineering college

Abstract— — Successful deployment of Electronic Health Record helps improve patient safety and quality of care, but it has theprerequisite of interoperability between Health Information Exchange at different hospitals. The Clinical Document Architecture (CDA) developed by HL7 is a core document standard to ensure such interoperability, and propagation of this document format is critical for interoperability. . A problem arises even when more hospitals start using the CDA document format because the data scattered in different documents are hard to manage. . Our CDA document integration system integrates multiple CDA documents per patient into a single CDA document and physicians and patients can browse the clinical data in chronological order.Cipher text-policy attribute-based encryption (CP-ABE) has been a preferred encryption technology to solve the challenging problem of secure data sharing in cloud computing. The shared data files generally have the characteristic of multilevel hierarchy, particularly in the area of healthcare and the military. However, the hierarchy structure of shared files has not been explored in CP-ABE. In this paper, an efficient file hierarchy attribute-based encryption scheme is proposed in cloud computing. The layered access structures are integrated into a single access structure, and then, the hierarchical files are encrypted with the integrated access structure. The ciphertext components related to attributes could be shared by the files. Therefore, both cipher text storage and time cost of encryption are saved. Moreover, the proposed scheme is proved to be secure under the standard assumption. Thus a secure and efficient privacy preserving dynamic medical text mining and image feature extraction scheme PPDM and file hierarchy in assisted E-health care systems is proposed.

Index Terms— Cloud computing, data sharing, file hierarchy, cipher text-policy, attribute-based encryption

1.INTRODUCTION

Electronic **H**ealth **R**ecord is longitudinal collection of health information for and about persons, wherehealth information is defined as information pertaining to the health of an individual or health care provided to an individual and it can support of efficient processes for health care delivery [1]. In order to ensure successful operation of EHR, a **H**ealth **I**nformation **E**xchange (HIE) system is needed in place [2]. However, most of the HIS in service are different and incompatible [3, 4]. Hence, effective health information exchange needs to be standardized for interoperable health information exchange between hospitals. Especially, clinical document standardization lies at the core of guaranteeing interoperability.

CDA (Clinical Document Architecture) by Health Level Seven is a major standard for clinical documents [5]. . The first version of CDA was developed in 2001 and Release 2 came out in 2005 [6]. CDA-based projects have been successfully completed in many countries [7-9]. Active works are being done on improving semantic interoperability based on openEHR and CEN13606 [10,11].

The adoption rate of EHR is very low except for a few handful countries such as New Zealand or Australia [12]. The US Government runs the Meaningful Use Program to improve efficiency in healthcare and patient safety. This program was launched as a part of incentives to raise the EHR adoption rate for EHR adopting hospitals [13].

The CDA document pertaining to a patient is generated at the clinic where the patient is diagnosed. The generated CDA document can be sent to other clinics after patient's consent is acquired. WITH the burgeoning of network technologyandmobile terminal, online data sharing hasbecomea new "pet", such as Face book, MySpace, and Badoo.Meanwhile, cloud computing [1]–[5] is one of the mostpromising application platforms to solve the explosive expanding of data sharing. In cloud computing to protect the data from leaking, users need to encrypt their data before being shared. Access control [6], [7] is paramount as it is the first line of defense that prevents unauthorized access to the shared data. Recently, attribute-based encryption (ABE) [8]–[10] has been attracted much more attentions since it can keep data privacy and realize fine-grained, one-to-many, and non-interactive access control. Cipher text-policy attribute based encryption (CP-ABE) [11]–[21] is one of feasible schemes which has much more flexibility and is more suitable for general applications [22],[23].

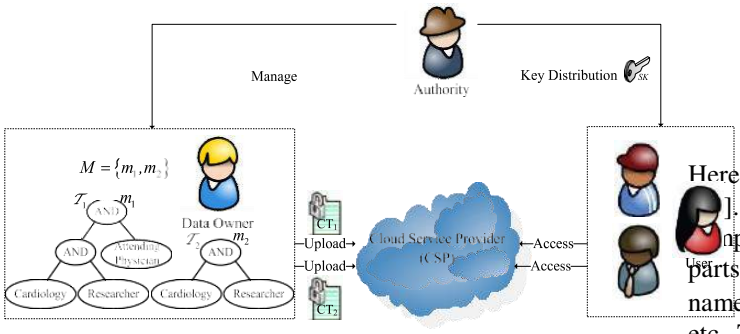


Fig 1. An example of secure data sharing in cloud computing

Incloudcomputing,asillustratedinFig.1,authorityaccepts the user enrollment and creates some parameters. Cloud ser- vice provider (CSP) is the manager of cloud servers and provides multiple services for client. Data owner encrypts and uploads the generated cipher textto CSP. User downloads and decrypts the interested cipher text from CSP. The shared files usually have hierarchical structure. That is, a group of files are divided into a number of hierarchy subgroups located at different access levels. If the files in the same hierarchical structure could be encrypted by an integrated access structure, thestoragecostofcipher textandtimecostofencryptioncould besaved.

Table:1: DATA ITEMS IN CCD HEADER AND SE TIONS IN THE CCD BODY	
CDA Header	Document information(creation time, template ID, language code, purpose)
	Patients information(ID,Name,Gender, birth date)
	Authors information(ID,Name,represented organization)
	Organization information(Name, Address, phone number)
CDA Body	Payers
	Advance Directives
	Support
	Function Status
	Problems
	Family History
	Social History
	Allergies
	Medications
	Medical Equipment
	Vital signs
	Results
	Procedures
	Encounters
	Plan of care

Here let us take the personal health record (PHR) for example]. To securely share the PHR information in cloud nputing, a patient divides his PHR information M into two parts: personal information$m1$ that may contain the patient's name, social security number, telephone number, home address, etc. The medical record $m2$ which does not contain sensitive personal information, such as medical test

results, treatment protocols, and operation notes. Then the patient adopts CP-ABE scheme to encrypt the information$m1$ and $m2$ by different access policies based on the actual need. For example, an attending physician needs to access both the patient's name and his medical record in order to make a diagnosis, and medical researcher only needs to access some medical test results for academic purpose inthe related area, where a doctor must be a medical researcher, and the converse is not necessarily true. Suppose that the patient sets the access structure of $m1$ as: $T1$ {("Cardiology" AND "Researcher") AND "Attending Physician"}. Similarly, $m2$ is termed as: $T2$ {"Cardiology" AND "Researcher"}. The example is deployed in cloud system as shown in Fig. 1. Apparently, the information needs to be encrypted twice if m1and m2are encrypted with access structures $T1$ and $T2$, respectively.

Twociphertexts$CT1 = \{T1, \tilde{C}1, C1, \forall y \in Y1 : Cy, C^{r}\}$ where $Y1$={"Cardiology", "Researcher","Attending","Physician"}and $CT2=\{T2, \tilde{C}2, C2, \forall y \in Y2: Cy, C^{r}\}$where $Y2$={"Cardiology", "Researcher"} will be produced [11]. The two structures could be integrated into one access structure T as shown in Fig 2.

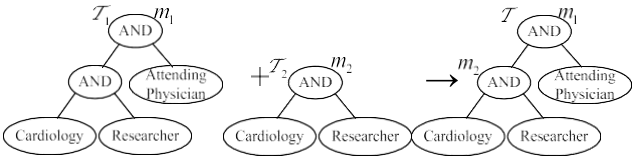
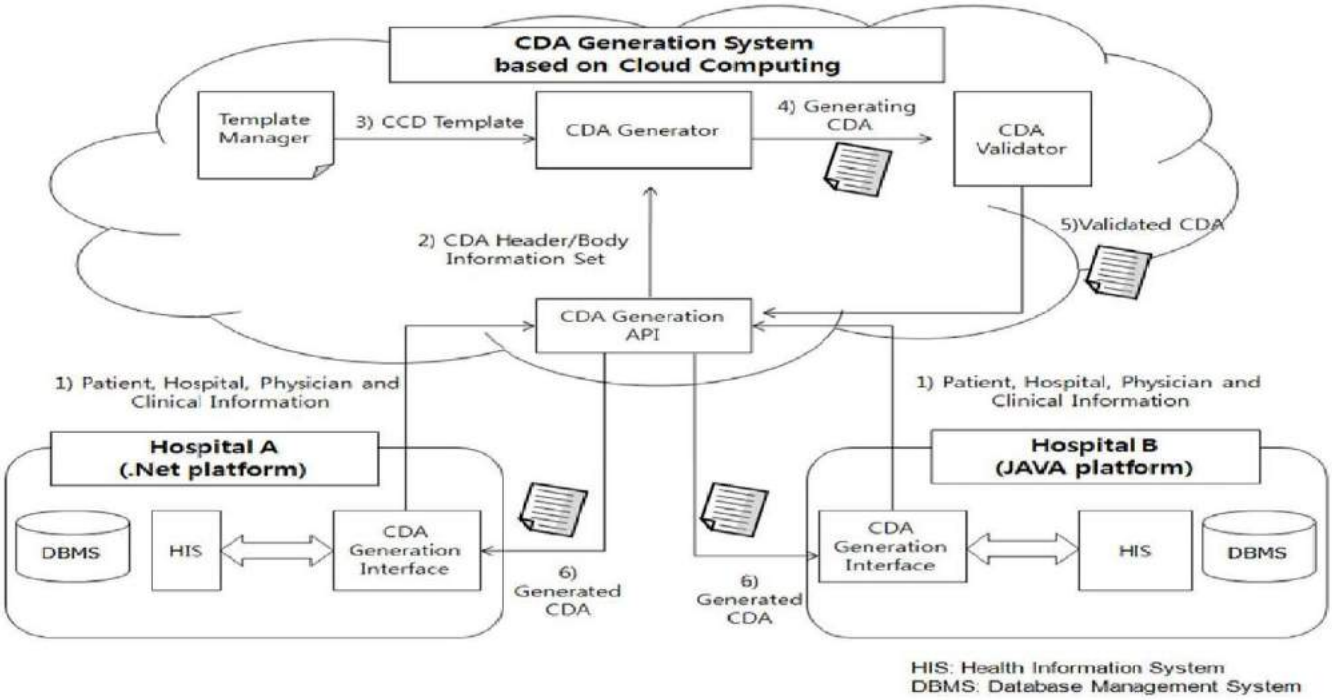


Fig. 2. The integrated access structure. $T1$ and $T2$ are access structures of $m1$ and $m2$, respectively. T is the integrated access structure of $m1$ and $m2$.

If the two files could be encrypted with the integrated access structure and produce cipher text $CT = \{T, \tilde{C}, C, \forall y \in Y : Cy, C^{r}\}$ where Y ={"Cardiology", "Researcher", "Attending Physician"}. Here, the components of cipher text $\{T, Cy, C^{r}\}$ are related to policy. Meanwhile, access structure could be shared

Fig. 1. Architecture of CDA generation system based in cloud computing



CDA Generation System Based on Cloud Computing

Fig. 1 shows the overall architecture o f how CDA documents can be generated on the health information systems of different hospital s by using our cloud computing-based CDA generation system.

Hospital A and Hospital B are demonstrated to show that it is easy to generate CDA documents on a variety of platforms if done via cloud. The purpose of each o f the components is:

- CDA Generation API generates CDA documents on cloud.

-CDA Generation Interface uses the API provided by the cloud and relays the input data and receives CDA documents generated in the cloud .

-Templates Manager manages the CDA documents to be generated in the cloud. This paper uses the CCD document template.

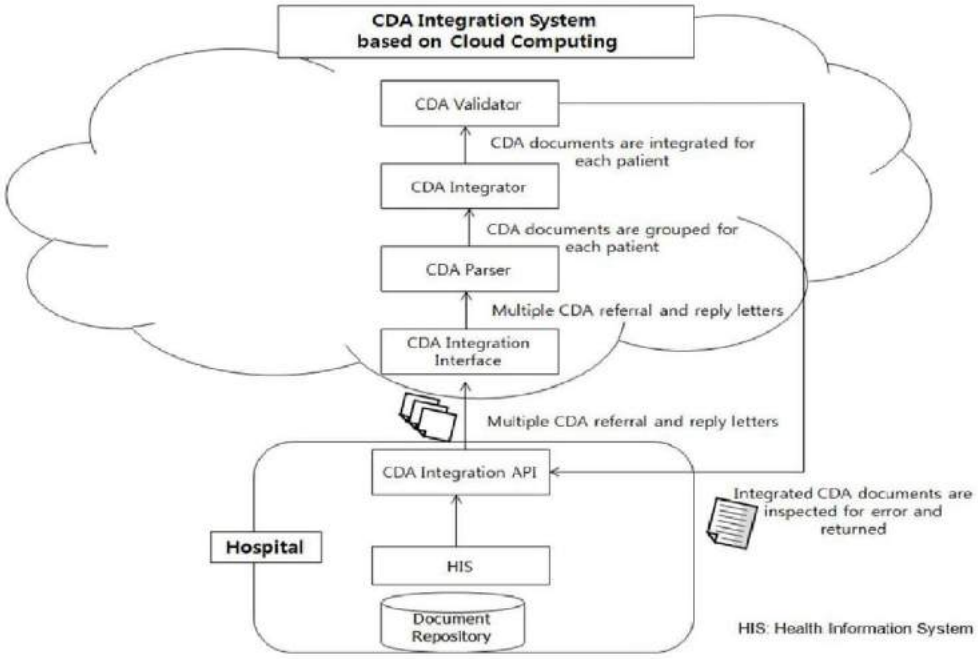
-CDA Generator uses the patient data received from different hospitals to generate CD A document s that fit in the format of templates from the Template Manager.

-C DA Validator inspects whether the generated CDA document complies with the CDA schema standards.

The DBMS at each hospital an d the HIS are linked as follows. Hospital A , which uses a .Net-based system is connected via ODB C to connect t o the DBMS while Hospital B , which uses a JAVA-base d system, is linked with Hibernate.

In a hospital, the clinical information of patient, hospital, and physician is entered via CDA Generation Inter-face a d sent to the cloud server via CDA Gene ration API. We utilize SOAP (Simple Object Access Protocol) astransmission protocol for the purpose of enhancing interoperability among different HIS when a hospital sends data t o the cloud. CDA Generation API relays the data in the CDA Header / Body in the list type. The items included in CDA Header are: PatientID, BirthDate, Gender, Given Name, and FamilyName; in CDA Body, the following items are included: Problem, Medication, Laboratory, Immunization, and so on. The data transmitted to the cloud server are put in CD A Header Se t and CDA Body Set and trans mitted to CD A Generator. CDA Generator retrieves a CCD template from Template Manager and fills in the appropriate fields of the CC D template with the data from the CDA Header / Body Sets. The gene rated CDA document is inspected by the CDA Validator whether the CDA standards are being satisfied. It is inspected whether there is any missing element or the for-mat is wrong here. If no error is found, a CDA document is returned to the recipient hospital. Hospitals A an d B are presented to demonstrate that it is possible for different development platform s to extend to generate C DA documents viacloud

Fig. 2. Architecture of CDA integration systembased on cloud computing.



Our Contributions

In this study, an efficient encryption scheme based on layered model of the access structure is proposed in cloud computing, which is named file hierarchy CP-ABE scheme (or FH-CP-ABE, for short). FH-CP-ABE extends typical CP-ABE [11] with a hierarchical structure of access policy, so as to achieve simple, flexible and fine-grained access control. The contributions of our scheme are three aspects.

- Firstly, we propose the layered model of access structure to solve the problem of multiple hierarchical files sharing. The files are encrypted with one integrated access structure.
- Secondly, we also formally prove the security of FH-CP-ABE scheme that can successfully resist chosen plaintext attacks (CPA) under the Decisional Bilinear Diffie-Hellman (DBDH) assumption.
- Thirdly, we conduct and implement comprehensive experiment for FH-CP-ABE scheme, and the simulation results show that FH-CP-ABE has low storage cost and computation complexity in terms of encryption and decryption.

It should be noticed that the proposed scheme differs fromthe subsequent CP-ABE schemes [26]–[30], which utilize the user layered model to distribute the work of key creation on multiple domain authorizations and lighten the burden of key authority center. In addition, the part of this work is presented in [25]. The work presented in that conference paper is rough and incomplete, where some important aspects haven't been considered.

A. Related Work

Sahai and Waters [8] proposed fuzzy Identity-Based

Encryption (IBE) in 2005, which was the prototype of ABE.

Latterly, a variant of ABE named CP-ABE [11], [12], [16], [31] was proposed.

In CP-ABE scheme, user's key SK is described by an attribute set S while cipher text CT is produced by encrypting plaintext M with an access policy T .

If data owner shares k files, i.e., $M = \{m_1, \ldots, m_k\}$,with same access policy T in the cloud, the files $M= \{m_1, \ldots, m_k\}$, can be encrypted and generated $CT = \{T, C\tilde{} = Me(g, g)^{\alpha s}, C = hs, \forall y \in Y : Cy = gqy(0), C`y = H(att(y))^{qy(0)}\}$ by using the typical CP-ABE scheme [11].If the k files $M = \{ m_1,\ldots,m_k\}$ have different access policies $T = \{T_1,\ldots,Tk\}$ in the cloud , the files can be encrypted individually and created cipher text $CT = \{CT1,\ldots,CTk\}$ by running CP-ABE scheme [11],where $CTi = \{Ti, C\tilde{} = m_i e(g,g)^{\alpha s}, C, \forall y \in Y : Cy, Cy`\}$. During the decryption, user can decrypt the ciphertext C Ti if and only if there is a "match" between the attributes set S and access structure Ti .

The work of key creation is distributed on multiple authorization domains and the burden of key authority center is lightened.

Other CP-ABE schemes with specific features have been presented. For example, Hur [38] proposed a data sharing schemetosolvetheproblemofkeyescrowbyusinganescrow free key issuing protocol between the key generation center and the data storing center. Green *et al.* [39] and Lai *et al.* [40] proposed CP-ABE schemes with outsourced decryption to reducetheworkloadofthedecryptionuser.AndFan*etal.*[22] proposed an arbitrary-state ABE scheme to solve the problem of the dynamic membership management. In addition, Guo *et al.* [15] proposed a novel constant-size decryption key CP-ABEschemeforstorage-constraineddevices.Hohenberger and Waters [41] proposed an online/offline ABE scheme to improve the speed of key generation and encryption, where each computation work in the two processes is split into two phases: offline phase (a preparation phase) and online phase.

C. Organization

The remaining parts of this paper are organized as follows. In section II, we introduce preliminaries which contain some notions and definitions. Then, the detailed construction and discussion of FH-CP-ABE scheme are presented in section III. In section IV, we provide security model and proof. The theoreticalanalysisandexperimentalsimulationswillbegiven in section V. Finally, the concluding remarks will be given in sectionVI.

I. PRELIMINARIES

In this section, notions used in this work are provided. More precisely, access structure, bilinear maps, DBDH assumption, and hierarchical access tree are introduced. The system definition and our basic construction are also presented.

A. AccessStructure

The description is similar as the literature [11]. Let $\{P_1, \ldots, P_n\}$ be a set of parties. A collection $\mathbb{A} \subseteq 2^{\{P_1,\ldots,P_n\}}$ is monotone if $\forall B, C$: if $B \in \mathbb{A}$ and $B \subseteq C$ then $C \in \mathbb{A}$. An access structure (respectively, monotone access structure) is a collection (respectively, monotone collection) $\mathbb{A}$ of non-empty subsets of $\{P_1, \ldots, P_n\}$, i.e., $\mathbb{A} \subseteq 2^{\{P_1,\ldots,P_n\}}\backslash\{\emptyset\}$. The sets in $\mathbb{A}$ are called the authorized sets, otherwise, the sets are called the unauthorized sets.

Generally, the data users are described by attributes. The authorized sets are included in $\mathbb{A}$. Unless stated in another way, the scheme uses an access structure which is a monotone form.

B. BilinearMaps

Let $\mathbb{G}_0$ and $\mathbb{G}_T$ be two groups of prime order p. The generator of $\mathbb{G}_0$ is g. A bilinear mapping $e : \mathbb{G}_0 \times \mathbb{G}_0 \rightarrow \mathbb{G}_T$ satisfies the following properties:

- Bilinearity: For any $u, v \in \mathbb{G}_0$ and $a, b \in \mathbb{Z}_p$, ithas $e(u^a, v^b) = e(u, v)^{ab}$.
- Non-degeneracy: There exists $u, v \in \mathbb{G}_0$ such that $e(u, v) \neq 1$.
- Computability: For all $u, v \in \mathbb{G}_0$, there is an efficient computation $e(u, v)$.

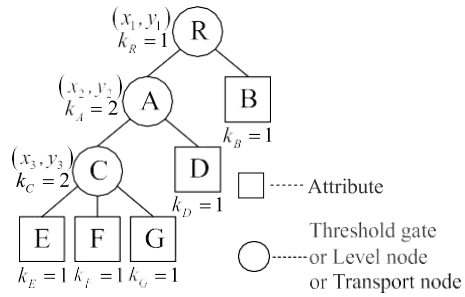


Fig. 3. An example of three-level access tree.

C. DBDH Assumption

A challenger chooses a group $\mathbb{G}_0$ of prime order p based on thesecurityparameterofsystem.Let$a, b, c \in \mathbb{Z}_p$ berandomly chosen and g be a generator of $\mathbb{G}_0$. With (g, g^a, g^b, g^c), the adversary must distinguish a valid tuple $e(g, g)^{abc} \in \mathbb{G}_T$ from a random element $R \in \mathbb{G}_T$.

An algorithm $\mathcal{B}$ that outputs a guess $\mu \in \{0, 1\}$ has advantage ε in solving DBDH in $\mathbb{G}_0$ if (1) was satisfied [31].

$$\left| Pr[\mathcal{B}(g, g^a, g^b, g^c, T = e(g,g)^{abc}) = 0] - Pr[\mathcal{B}(g, g^a, g^b, g^c, T = R) = 0] \right| \geq \varepsilon \quad (1)$$

Definition 1: We say that the DBDH assumption holds if no polynomial algorithm has a non-negligible advantage in solving the DBDHproblem.

D. Hierarchical AccessTree

Let $\mathcal{T}$ be a hierarchical tree representing an access structure which is divided into k access levels. Nodes of the tree are denoted as (x, y). The symbol x represents the node's row in $\mathcal{T}$ (from top to bottom), and y represents the node's column in $\mathcal{T}$ (from left to right). In Fig. 3, the nodes can be denoted as: R = (1, 1), A = (2, 1), B = (2, 2), C = (3, 1), D = (3, 2), E = (4, 1), F = (4, 2), G = (4, 3). To facilitate description of the access tree, several functions and terms are defined as follows.

- (x, y). It denotes a node of tree $\mathcal{T}$. If (x, y) is a leaf node, it denotes an attribute. If (x, y) is a non-leaf node, it denotes a threshold gate, such as "AND", "OR", "n-of-m (n<m)". For example, the nodes A and E denote a threshold gate and an attribute in Fig. 3.
- $num_{(x,y)}$. It denotes the number of (x, y)'s children in $\mathcal{T}$. For example, $num_R = 2$ in Fig. 3.
- $k_{(x,y)}$. It denotes threshold value of node (x, y),where $0 < k_{(x,y)} \leq num_{(x,y)}$. When $k_{(x,y)} = 1$ and (x, y) is a non-leaf node, (x, y) is an OR gate. When $k_{(x,y)} = num_{(x,y)}$ and (x, y) is a non-leaf node, it is an AND gate. In particular, if (x, y) is a leaf node, $k_{(x,y)} = 1$. For example, $k_A = 2$ denotes an AND gate in Fig.3.
- $(x_i, y_i)(i \in [1, k])$. It denotes level node of $\mathcal{T}$. In this work, access tree $\mathcal{T}$ is divided into k access levels. And thehierarchyofthenodesissortedindescendingorder. That is, (x_1, y_1) is the highest hierarchy, and $(x_k, y_k)$is

the lowest hierarchy. For example, (x_2, y_2) = A is the second hierarchy in Fig. 3.

- $parent(x,y)$. It represents the parent of the node (x,y) in $\mathcal{T}$. For example, $parent(2, 1) = parent(A) = R$ in Fig.3.
- transport node. The node (x , y) is a transport node if one of the children of (x , y) contains at least one threshold gate. For example, R and A are transport nodes in Fig.3.
- $TN - CT(x , y)$. It represents a threshold gate set of transport node (x , y)'s children in $\mathcal{T}$. It is marked as $TN - CT(x , y) = \{child_1, child_2, .. .\}$. Forexample, $TN - CT(A) = \{C\}$ in Fig.3.
- $att(x,y)$. It denotes an attribute associated with the leaf node (x , y) in $\mathcal{T}$.
- $index(x,y)$. It returns a unique value associated with the node (x,y), where the value is assigned to (x,y) for a given key in an arbitrary manner.
- $\mathcal{T}_R$. It denotes a tree diagram, where root node of the tree is R.
- $\mathcal{T}_{(x ,y)}$. It denotes the sub-tree of $\mathcal{T}$ rooted at the node (x,y). If an attribute set S satisfies $\mathcal{T}_{(x,y)}$, we denote it as $\mathcal{T}_{(x ,y)}(S) = 1$. $\mathcal{T}_{(x ,y)}(S)$ is recursively computed as follows. If (x , y) is a non-leaf node, $\mathcal{T}_{(x ,y)}(S)$ returns 1 if and only if at least $k_{(x ,y)}$ children return 1. If (x , y) is a leaf node, then $\mathcal{T}_{(x ,y)}(S)$ returns 1 if and only if $att(x,y) \in S$.

E. System Definition and Our Basic Construction

As illustrated in Fig. 1, the system model in cloud computing is given, which consists of four different entities: authority, CSP, data owner and user. In this work, we assume that data owner has k files with k access levels and $M = \{m_1, \ldots , m_k\}$ is shared in cloud computing. Here, m_1 is the highest hierarchy and m_k is the lowest hierarchy. If a user can decrypt m_1, the user can also decrypt $m_2, \ldots, m_k$.

Authority: It is a completely trusted entity and accepts the user enrollment in cloud computing. And it can also execute **Setup** and **KeyGen** operations of the proposed scheme.

Cloud Service Provider (CSP): It is a semi-trusted entity in cloud system. It can honestly perform the assigned tasks and return correct results. However, it would like to find out as much sensitive contents as possible. In the proposed system, it provides ciphertext storage and transmission services.

Data Owner: It has large data needed to be stored and shared in cloud system. In our scheme, the entity is in charge of defining access structure and executing **Encrypt** operation. And it uploads ciphertext to CSP.

User: It wants to access a large number of data in cloud system. The entity first downloads the corresponding ciphertext. Then it executes **Decrypt** operation of the proposed scheme.

In Fig. 4, a data owner processes the files as follows: Firstly, the data owner chooses k content keys $\{ck_1, \ldots , ck_k\}$, and encrypts files $\{m_1, \ldots , m_k\}$ with the content keys by using symmetric encryption algorithm (i.e., DES, AES). The ciphertexts are denoted as $E_{ck}(M) = \{E_{ck_1}(m_1), .. ., E_{ck_k}(m_k)\}$. Then, the data owner encrypts $\{ck_1, \ldots , ck_k\}$

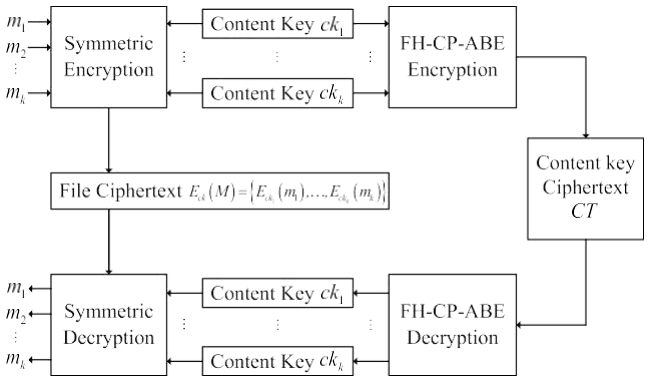


Fig. 4. The system framework of FH-CP-ABE scheme.

using FH-CP-ABE encryption algorithm and obtains an integrated ciphertext of content keys CT.

The procedures of decryption is described as below. Firstly, the user decrypts ciphertext CT and obtains content key by using FH-CP-ABE decryption operation. Then, the user can obtain file by using symmetric decryption algorithm with content key.

Definition 2: The FH-CP-ABE scheme consists of four operations: **Setup**, **KeyGen**, **Encrypt** *and* **Decrypt**. *It is described as follows:*

1) $(PK, MSK) \leftarrow$ **Setup**(1^κ). The probabilistic operation takes a security parameter κ as input and outputs public key PK and master secret key MSK.
2) $(SK) \leftarrow$ **KeyGen**(PK, MSK, S). The operation inputs PK, MSK and a set of attributes S and creates a secret key SK.
3) $(CT) \leftarrow$ **Encrypt**(PK , ck, A). The operation inputs PK, $ck = \{ck_1, \ldots, ck_k\}$ and a hierarchical access tree A as shown in the Fig. 2. At last, it creates an integrated ciphertext of content keys CT.
4) $(ck_i (i \in [1,k])) \leftarrow$ **Decrypt**(PK, CT, SK). The algorithm inputs PK , CT which includes an inte- grated access structure A, SK described by a set of attributes S. If the S matches part of A, some content keys ck_i $(i \in [1, k])$ can be decrypted. If it matches the whole A, all the content keys can be decrypted. Then, the corresponding files m_i $(i \in [1, k])$ will be decrypted with the content keys by the symmetric decryption algorithm.

In addition, an example of FH-CP-ABE scheme in cloud computing is shown in Fig. 5. Suppose that a data owner wants to share two hierarchical files m_1 and m_2. It encrypts the two files with the randomly chosen content keys ck_1 and ck_2, and generates ciphertext $\{E_{ck_1}(m_1), E_{ck_2}(m_2)\}$. Then, based on the hierarchical access tree, the content keys ck_1 and ck_2 are encrypted as follows: Firstly, authority generates public key PK and master secret key MSK by **Setup** operation of FH-CP-ABE scheme. Secondly, authority creates secret key SK for each user by **KeyGen** operation. Thirdly, data owner encrypts content keys $ck = \{ck_1, ck_2\}$ under the access policy $\mathcal{T}$. It creates one integrated CT by **Encrypt** operation and uploads the ciphertext to CSP. Finally, if a user

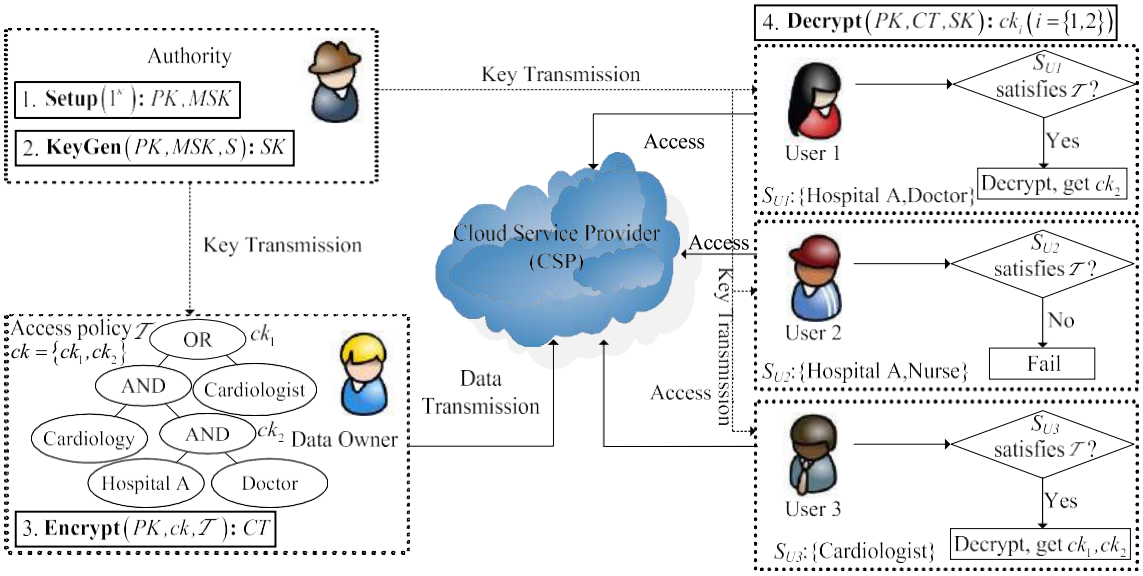


Fig. 5. An example of FH-CP-ABE scheme used in cloud computing. Data owner encrypts content keys $ck = \{ck_1, ck_2\}$ under the access policy $\mathcal{T}$. Users decrypt some or all content keys if users' attribute set satisfies part or the whole $\mathcal{T}$.

wants to access cloud files, it first downloads the ciphertext $\{E_{ck_1}(m_1), E_{ck_2}(m_2)\}$ and CT from CSP. Then it can decrypt some or all content keys if and only if the user's attribute set satisfies part or the whole $\mathcal{T}$. As shown the Fig. 5, the user 1 can decrypt ck_2 to further obtain m_2. The user 2 cannot decrypt any message, and the user 3 can decrypt ck_1 and ck_2 to further decrypt m_1 and m_2.

II. The Proposed FH-CP-ABE Scheme

In this section, the detailed construction of FH-CP-ABE scheme is first presented. Then, based on the scheme, an improved encryption process about FH-CP-ABE scheme is proposed in order to reduce computational complexity. In addition, a brief discussion about FH-CP-ABE scheme's features is alsoprovided.

A. Scheme Construction

Let $e : G_0 \times G_0 \rightarrow G_T$ be a bilinear map, and G_0 be bilinear group of prime order p with generator g. For any $k \in Z_p$ and an attribute set $S = \{S_1, S_2, \ldots, S_m \in Z_p\}$, the Lagrange coefficient $\delta_{k,S} = \prod_{l \in S, l \neq k} (x - l)/(k - l)$. Two hash functions $H_1 : \{0,1\}^* \rightarrow G_0$ and $H_2 : \{0,1\}^* \rightarrow G_T$ are used in the proposed scheme. An universe of attribute set is defined as $\tilde{A} = \{a_1, \ldots, a_n\}$.

1) **Setup**(1^κ). The authority runs the operation which inputs a security parameter κ and chooses random numbers $\alpha, \beta \in Z_p$. It outputs PK and MSK as the formulas (2) and (3), respectively.

$$PK = \{G_0, g, h = g^\beta, e(g,g)^\alpha\} \quad (2)$$

$$MSK = \{g^\alpha, \beta\} \quad (3)$$

2) **KeyGen**(PK, MSK, S). The authority executes the algorithm which inputs a set of attributes $S (S \subseteq \tilde{A})$ and creates a secret key SK about the set as the formula (4), where $r \in Z_p$ and $r_j \in Z_p$ are randomly chosen for each user and each attribute $j \in S$.

$$SK = \begin{matrix} D = g^\alpha \cdot h^r, \\ \forall j \in S: D_j = g^r \end{matrix} \cdot H_1(j)^{r_j}, D'_j = h^{r_j} \quad (4)$$

3) Assume that a data owner shares k files, i.e., $M = \{m_1, \ldots, m_k\}$, with k access levels. Then, the corresponding content keys $ck = \{ck_1, \ldots, ck_k\}$ are encrypted as the following **Encrypt** operation.

Encrypt$(PK, ck, \mathcal{T})$. The public key PK, content keys $ck = \{ck_1, \ldots, ck_k\}$, and a hierarchical access tree $\mathcal{T}$ are taken as input. The algorithm outputs an integrated ciphertext CT.

- Data owner sets level nodes $(x_i, y_i)(i = 1, 2, \ldots, k)$ in $\mathcal{T}$, and selects k random numbers $s_1, \ldots, s_k$ in Z_p. Then, it computes $\dot{C}_i$ and C'_i for all $i = 1, 2, \ldots, k$ as the formula (5).

$$\dot{C}_i = ck_i e(g,g)^{\alpha s_i}, C'_i = g^{s_i} \quad (5)$$

- Polynomial structure rule: a polynomial $q_{(x,y)}$ needs to be selected for each node (x, y) (including the leaf nodes) in $\mathcal{T}$. From the root node R, the nodes' information of $q_{(x,y)}$ is randomly selected from top to bottom manner. For each node (x, y) in $\mathcal{T}$, degree of the polynomial $d_{(x,y)}$ is set to $k_{(x,y)} - 1$, where $k_{(x,y)}$ is the threshold value.
- Beginning from the root node R, data owner sets $q_R(0) = q_{(x_1,y_1)}(0) = s_1$ and chooses d_R other points of the polynomial q_R to define it completely, where the points are made of two types of nodes. The one are level nodes which are children of R. The other are remaining nodes randomly selected. For each non-root node (x, y), it sets $q_{(x,y)}(0) = q_{(x_i,y_i)}(0) = s_i$ if the (x, y) is a level node. Otherwise, $q_{(x,y)}(0) = q_{parent(x,y)}(index(x, y))$.

The other $d_{(x,y)}$ points of $q_{(x,y)}$ are made of the level nodes of the children of (x, y) and the remaining nodes randomly selected.

Let Y be the set of leaf nodes in T. Then, data owner computes $C_{(x,y)}$ and $C'_{(x,y)}$ for all nodes (x, y) in the set of Y as the formulas (6) and (7).

$$C_{(x,y)} = h^{q_{(x,y)}(0)}, \quad (6)$$

$$C'_{(x,y)} = H_1(att(x,y))^{q_{(x,y)}(0)} \quad (7)$$

- In T, let X be the set of transport nodes, and $TN - CT_{(x,y)}$ be the threshold gate set of transport node (x, y)'s children, where $TN - CT(x, y) = \{child_1, \ldots, child_j, \ldots\}$. Then, data owner computes $\hat{C}_{(x,y),j}$ for each node (x, y) in the set of X and all $j = 1, 2, \ldots$ as the formula (8).

$$\hat{C}_{(x,y),j} = e(g,g)^{\alpha \cdot (q_{(x,y)}(0) + q_{child_j}(0))} \cdot H_2(e(g,g)^{\alpha q_{(x,y)}(0)}) \quad (8)$$

- Data owner outputs the integrated ciphertext CT as the formula(9).

$$CT = \{T, \tilde{C}_i, C', C_{(x,y)}, C'_{(x,y)}, \hat{C}_{(x,y),j}\} \quad (9)$$

of leaf nodes is $Y = \{\text{“}Y_1\text{”}, \text{“}Y_2\text{”}, \text{“}Y_3\text{”}\} = \{\text{“Cardiology”}, \text{“Researcher”}, \text{“AttendingPhysician”}\}$.

4) **Decrypt** (PK, CT, SK). A user needs the public key PK and SK described by S to decrypt CT. Similar to CP-ABE [11], a recursive operation $DecryptNode(CT, SK, (x,y))$ should be first defined.

(1) If (x, y) is a leaf node, we let $i = att(x, y)$

If $i \notin S$, $DecryptNode(CT, SK, (x,y)) = null$.

Otherwise, the operation $DecryptNode(CT, SK, (x, y))$ is obtained by the formula of(10)

$$Decrypt = \frac{e(g^r H_1(i)^{r_i}, h^{q_{(x,y)}(0)})}{e(h^{r_i}, H_1(att(x, y))^{q_{(x,y)}(0)})} = e(g,g)^{r \beta q_{(x,y)}(0)} \quad (10)$$

(2) If (x, y) is a non-leaf $(CT, SK, (x,y))$ is defined as below. For all nodes z that are children of (x,y), it runs $DecryptNode(CT, SK, z)$ and stores the output as F_z. Let $S_{(x,y)}$ be an arbitrary $k_{(x,y)} - sized$ child nodes set z, and then $F_z \neq null$. If the set does not exist, $F_z = null$. Otherwise, $F_{(x,y)}$ is computed as the formula (11),

where $S'_{(x,y)} = \{index(z) : z \in S_{(x,y)}\}$, $i = index(z)$.

$$F_{(x,y)} = \prod_{z \in S_{(x,y)}} F_z^{\Delta_{i,S'_{(x,y)}}(0)} = \prod_{z \in S_{(x,y)}} (e(g,g)^{r \cdot \beta q_z(0)})^{\Delta_{i,S'_{(x,y)}}(0)} = \prod_{z \in S_{(x,y)}} e(g,g)^{r \cdot \beta q_{(x,y)}(i) \Delta}$$

Then, the procedures of decryption algorithm are Described as follows.

- If the attribute set S satisfies part or the whole T, that is, S satisfies part or the whole level nodes, $e(g, g)^{r \beta s_i} (i \in [1, k])$ can be obtained by the recursive operation of the formula (12).

$$A_i = DecryptNode(CT, SK, (x,y))_i = e(g,g)^{r \beta q_{(x_i, y_i)}(0)} = e(g,g)^{r \beta s_i} (i \in [1,k]) \quad (12)$$

- Based on the hierarchical nodes, if S includes the lower authorization nodes, we can recursively calculate all of the authorization's level nodes with the values of transport nodes $\hat{C}_{(x,y),j}$ $(j = 1,2,\ldots)$ by using the formula(14).

Therefore,

$F_{(i+1),j}, \ldots, F_{k,j}$ are obtained sequentially. That is, the values $e(g, g)^{\alpha s_i}, e(g,g)^{\alpha s_{i+1}}, \ldots, e(g,g)^{\alpha s_k}$ equation that has the decrypt method of j value.

j value of 1,2,3,……..n.

$$node = e(g,g)^{\alpha (j = 1,2,\ldots)} \quad (14)$$

- Then, the corresponding content keys $\{ck_i, \ldots, ck_k\}$ are decrypted by executing the formula (15) repeatedly.

$$\frac{\tilde{C}_i}{F_i} = \frac{ck_i e(g, g)^{\alpha s_i}}{e(g, g)^{\alpha s_i}} = ck_i \ (i \in [1,k]) \quad (15)$$

- At last, the authorized files $\{m_i, \ldots, m_k\}$ are decrypted with $\{ck_i, \ldots, ck_k\}$, using symmetric decryption algorithm.

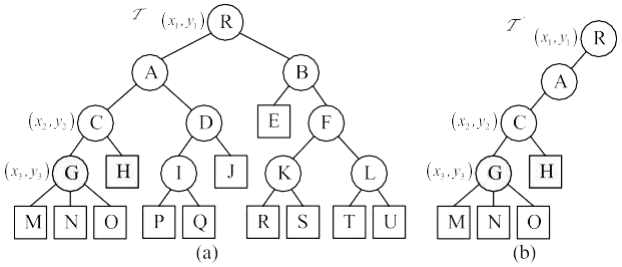


Fig.6. A three-level access tree. (a) The general access tree $\mathcal{T}$.(b)Thesimplified access tree $\mathcal{T}^r$. Each leaf node of access tree denotes an attribute.
Each non-leaf node of access tree is a threshold gate.

B. FH-CP-ABE Scheme With ImprovedEncryption

To facilitate the presentation in the below, we denote the above FH-CP-ABE scheme as BasicFH-CP-ABE. We now show how to modify the encryption process of BasicFH-CP-ABE scheme in order to reduce computational complexity. In ciphertext CT , some transport nodes are removed from CT if they don't carry any information about level node, where the information denotes leaf node, non-leaf node, level node, or transport node in hierarchical access tree. That is, these transport nodes are removed from CT if they do not directly or indirectly contain level node. More precisely, we improve the part $\hat{C}_{(x,y),j}$ about transport node in CT . All other operations execute exactly as in BasicFH-CP-ABE. In order to make a clear description, we use an example to further illustrate the improved encryption process in the step of $\hat{C}_{(x,y),j}$ of ciphertext.

Assume that data owner defines a three levels access tree $\mathcal{T}$ as shown in Fig. 6(a). In the $\mathcal{T}$, we get the following information. The attribute set is {E, H, J, M, N, O, P, Q, R, S, T, U}. On the contrary, the set of threshold gate denotes {R,A,B,C,D,F,G,I,K,L}. The set of transport nodes is {R, A, B, C, D, F}, where $TN-CT(R)=\{A, B\}$, $TN-CT(A)=\{C,D\}$,$TN-CT(B)=\{F\}$,$TN-CT(C)=\{G\}$, $TN-CT(D)=\{I\}$, $TN-CT(F)=\{K, L\}$.

In the phase of **Encrypt** of BasicFH-CP-ABE, there are 9 eligible children threshold gates related to transport nodes in $\mathcal{T}$ because the set $TN-CT(x, y)$ contains 9 children thresholdgates.So,the part$\hat{C}_{(x,y),j}$inciphertextiscomputed 9-times. However, we find that some calculations are not needed as the transport nodes don't carry any information about level node, such as the nodes B, D, and F. To improve this aspect, the access tree $\mathcal{T}$ is simplified to the $\mathcal{T}^r$ as illustrated in Fig 6(b), where the simplified rule of access tree is described as below. In $\mathcal{T}^r$, the number of eligible children threshold gates related to transport nodes is reduced to 3, where $TN-CT(R)=\{A\}$, $TN-CT(A)=\{C\}$, $TN-CT(C)=\{G\}$in$\mathcal{T}^r$.The part$\hat{C}_{(x,y),j}$inciphertext is computed 3-times instead of the original 9-times in G_T, thus both the computational complexity and storage cost of this work arereduced.

- The simplified rule of access tree: in a general access tree (as shown in the Fig 6(a)), the transport node corresponding sub-tree should be erased if the transport node is not level node and all of the children nodes of the transport node don't contain level node, where the reason isthat these transport nodes do not carry any information about level node. From the root node, the general access tree is simplified from top to bottom manner. The simplified rule is applied in the step of producing $\hat{C}_{(x,y),j}$ in the encryption and decryption processes, respectively.

C. SchemeDiscussion

We now provide a brief discussion about FH-CP-ABE scheme's features for the entities data owner and user. Here we suppose that data owner needs to share k hierarchical files with k access levels in cloud computing as we set before.

1) *Computation Cost on Data Owner.* In the proposed scheme, the layered model of access structure is provided so as to achieve multiple hierarchical files sharing. The files are encrypted with one integrated accessstructure.So,dataownercanencryptthedifferent levels of the files and generate an integrated ciphertext only executing the encryption algorithm one time. Especially,somecommonattributesshouldbecomputed only once instead of many times since each common attribute is appeared in the integrated access structure one time, where the common attribute denotes that it is appeared in multiple access sub-trees associated with k hierarchical files. Assume that there are n common attributes appeared in the $m(m \le k)$ access sub-trees associated with the k hierarchical files, the computation workload of the common attributes about ciphertext $C_{(x,y)}$ and $C'_{(x,y)}$ (as shown in the formulas (6) and (7)) is reduced by $2n(m-1/m)$, thus the encryption efficiency of data owner is also improved.
2) *Computation Cost on User.* In decryption process, users can decrypt all of his authorization files with computation of secret key once since transport nodes are added in the access structure with k level nodes. And the bilinear pairing operation of each common nodes used in multiple access sub-trees of the files is computed only once since each common node is alsoappeared in the integrated access structure one time. Assume that there are n common nodes appeared in the $m(m \le k)$ access sub-trees associated with the k hierarchical files, the bilinear pairing computation cost of the common nodesaboutdecryption$DecryptNode(CT,SK,(x,y))$ (as shown in the formula (10)) is also reduced by $2n(m-1/m)$ if the user needs to decrypt the k files.

III. SECURITYANALYSIS

Security of this work involves two aspects: file ciphertext confidentiality and content key ciphertext confidentiality. We assume that the hierarchical files are safely encrypted by using symmetric encryption algorithm (i.e., DES, AES). Therefore only the security proof of FH-CP-ABE should be provided. In this section, the security game of the proposed scheme is given firstly. Then a formal security proof is provided based on the results of the literatures [12], [31].

A. CPA Security Game for the Proposed Scheme

In the proposed scheme, SK is user's secret key associated with attribute set, and CT denotes ciphertext associated with

access structure. Security model of the proposed scheme

restriction is that SK does not satisfy $\mathbb{A}^*$.

1) *Initialization*. The adversary A selects the challenging access structure $\mathbb{A}^*$ and submits $\mathbb{A}^*$ to the challenger C.
2) *Setup*. C runs the **Setup** operation of FH-CP-ABE scheme and sends public key PK to A.
3) *Query Phase* 1. For the attribute sets $S_1, \ldots, S_{q1}$ ($\forall i \in [1, \ldots, q1]$, $S_i \notin \mathbb{A}^*$) chosen by A, he can repeatedly ask C for the secret keys SK. Meanwhile, C answers these secret keys SK by running **KeyGen** algorithm.
4) *Challenge*. A submits two messages m_0 and m_1 of equal length. C randomly picks a bit $\mu \in \{0, 1\}$ and encrypts m_μ with $\mathbb{A}^*$. The resulting ciphertext CT^* is given to A.
5) *Query Phase* 2. Same as the *Query Phase* 1.
6) *Guess*. Finally, A guesses $\hat{\mu} \in \{0,1\}$. If $\hat{\mu} = \mu$, A wins the security game. In this game, A can win the game which is defined as $Adv(1^\kappa)_A = |Pr[\hat{\mu} = \mu] - (1/2)|$.

Definition 3: The proposed scheme is said to be secure against CPA if no probabilistic polynomial-time adversaries have non-negligible advantage in the above game.

B. Security Proof for the Proposed Scheme

Theorem 1: Suppose DBDH assumption holds. Then no polynomial adversary can selectively break the proposed system.

Proof: Suppose that the adversary A has non-negligible advantage $\varepsilon = Adv_A$ in the selective security game against our construction. Then, we construct a simulator B that can distinguish a DBDH tuple from a random tuple with advantage $\frac{\varepsilon}{2}$. Let $e : G_0 \times G_0 \rightarrow G_T$ be an efficiently computable bilinear map, where G_0 has prime order p with generator g. First the DBDH challenger randomly selects the following parameters. $a, b, c \in Z_p, \mu \in \{0, 1\}$, generator $g \in G_0$ and a random element $R \in G_T$. The challenger defines T to be $e(g, g)^{abc}$ if $\mu = 0$. Otherwise, he sets $T = R$. He then gives the simulator $(g, A, B, C, T) = (g, g^a, g^b, g^c, T)$. The simulator B now plays the role of challenger in the security game. In order to make the description clearly, only one file is encrypted. Therefore, the ciphertext CT can be simplified as formula (9).

1) *Initialization*. The adversary A selects the challenging access structure $\mathbb{A}^*$ and sends $\mathbb{A}^*$ to the simulator B.
2) *Setup*. To provide a public key PK to A, B randomly chooses a number $\alpha' \in Z_p$, and notes $\alpha = \alpha' + ab$. It computes $e(g, g)^\alpha$ like this: $e(g, g)^\alpha = e(g,g)^{\alpha'} e(g,g)^{ab}$. Meanwhile, it sets $h = g^\beta = B = g^b$. At last, B gives PK to A.
3) *Query Phase* 1. In this phase, A can query the secret key SK by submitting an attribute set $W_j = \{a_j | a_j \in \mathbb{A}\}$ ($W_j \notin \mathbb{A}^*$) to B. Firstly, B randomly picks a number $r' \in Z_p$, and sets $r = r' - a$. It can obtain $D = g^\alpha \cdot h^r = g^{a,} g^{\beta r} = g^{\alpha' + ab} \cdot g^{b(r' - a)} = g^{(\alpha' + r'b)}$. Then, for each attribute $a_j \in W_j$, B needs to randomly choose $r_j \in Z_p$. It constructs the remaining secret key as follows:

4) *Challenge*. Eventually submits two messages bit $\hat{\mu} \in \{0,1\}$. By running encryption operation under $\mathbb{A}^*$, B computes CT^* as $C' = g^s = g^c = C$, $\tilde{C} = m_{\hat{\mu}} \cdot e(g,g)^{as} = m_\mu \cdot e(g,g)^{ac} = m_\mu \cdot Te(g,g)^{a'c}$. Finally, B sends CT^* to A.
5) *Query Phase* 2. Same as the *Query Phase* 1.
6) *Guess*. Finally, A outputs a guess $\hat{\mu}$ of $\hat{\mu}$. If $\hat{\mu} = \hat{\mu}$, B outputs 0 to guess that $T = e(g, g)^{abc}$. Otherwise, it outputs 1 to indicate that T is a random group element in G_T.

If $T = e(g, g)^{abc}$, then CT is a valid ciphertext, in which case the advantage of the adversary is ε.

$$Pr[B(g, g^a, g^b, g^c, T = e(g,g)^{abc}) = 0] = \frac{1}{2} + \varepsilon \quad (16)$$

If $T = R$, then $\tilde{C}$ is completely random from the view of the adversary. Therefore $\hat{\mu}' = \hat{\mu}$ holds with probability exactly $\frac{1}{2}$, regardless of the distribution on $\hat{\mu}$.

$$Pr[B(g, g^a, g^b, g^c, T = R) = 0] = \frac{1}{2} \quad (17)$$

Lastly, the advantage of the simulator in this security game is described as follows.

$$Adv_S = \frac{1}{2} Pr[B(g, g^a, g^b, g^c, T = e(g,g)^{abc}) = 0] + \frac{1}{2} Pr[B(g, g^a, g^b, g^c, T = R) = 0] - \frac{1}{2}$$

IV. Performance Analysis

In this section, the results of theoretical analysis and experimental simulation are given. The experimental results show that the proposed scheme is highly efficient, particularly in

A. Theoretical Analysis

The notations used in theoretical analysis are firstly defined. Let $G_i (i = 0, T)$ be the group or operation in group (exponentiation, multiplication), and Z_p be the group $\{0, 1, \ldots, p-1\}$ with multiplication modulo p, where p is a prime. Let C_e be the e operation, and e denotes bilinear pairing. Let A_u be the attributes of user u, and S be the least interior nodes satisfying an access structure (include the root). Let A_C be the attributes with ciphertext CT and A_T be the set of transport nodes. Let L_* be bit-length of element in $*$, and $|*|$ be the number of elements in $*$. In addition, the number of the nodes in the set $TN - CT(x, y)$ is considered in the following analysis,

TABLE I

COMPARISONS OF THE FEATURES OF CP-ABE WITH FH-CP-ABE ($M = \{m_1, \ldots, m_k\}$)

Component	CP-ABE	FH-CP-ABE
Encryption Time	$[2(\lvert\mathbb{A}_{C1}\rvert + \ldots + \lvert\mathbb{A}_{Ck}\rvert) + k]\mathbb{G}_0 + 2k\mathbb{G}_T$	$(2\lvert\mathbb{A}_{C1}\rvert + k)\mathbb{G}_0 + (2\lvert\mathbb{A}_T\rvert + 2k)\mathbb{G}_I$
Decryption Time	$k(2\lvert\mathbb{A}_u\rvert + 1)C_e + [2(\lvert S_1\rvert + \ldots + \lvert S_k\rvert) + 2k]\mathbb{G}_T$	$(2\lvert\mathbb{A}_u\rvert + 1)C_e + [2\lvert S\rvert + (\lvert\mathbb{A}_T\rvert + 2k)]\mathbb{G}_I$
The Size of PK	$3L_{\mathbb{G}_0} + L_{\mathbb{G}_T}$	$3L_{\mathbb{G}_0} + L_{\mathbb{G}_T}$
The Size of MSK	$L_{\mathbb{Z}_p} + L_{\mathbb{G}_0}$	$L_{\mathbb{Z}_p} + L_{\mathbb{G}_0}$
The Size of SK	$(2\lvert\mathbb{A}_u\rvert + 1)L_{\mathbb{G}_0}$	$(2\lvert\mathbb{A}_u\rvert + 1)L_{\mathbb{G}_0}$
The Size of CT	$[2(\lvert\mathbb{A}_{C1}\rvert + \ldots + \lvert\mathbb{A}_{Ck}\rvert) + k]L_{\mathbb{G}_0} + kL_{\mathbb{G}_T}$	$(2\lvert\mathbb{A}_{C1}\rvert + k)L_{\mathbb{G}_0} + (\lvert\mathbb{A}_T\rvert + k)L_{\mathbb{G}_T}$

as the nodes are closely related to the computation and storage cost of transport node which is associated with encryption, decryption, and ciphertext CT (as shown in the formulas (8) and (14)). To facilitate the analysis, we assume that the set $TN - CT(x, y)$ of each transport node (x, y) contains j nodes, that is, $TN-CT(x,y)=\{child_1, \ldots, child_j\}$.

Suppose that there are k hierarchical files, i.e., $M = \{m_1, \ldots, m_k\}$, and their access order is decreased. Thus, the attributes can be denoted as $\{\mathsf{A}_{C1}, \ldots, \mathsf{A}_{Ck}\}$, where $\mathsf{A}_{C1} \supset \mathsf{A}_{C2} \supset \ldots \supset \mathsf{A}_{Ck}$. Similar to Fig. 2, there is an access structure which contains k hierarchical nodes. The least interior nodes set are denoted as $\{S_1, \ldots, S_k\}$. The features comparison between proposed scheme and CP-ABE [11]is listed on Table I, where access structure T and simple hash computation are not included.

As illustrated in Table I, when j contained in $TN-CT(x,y)$ is fixed, we can observe that the computational time of encryption increases linearly with the number of documents k, and the growth rate is equal to $(\mathsf{G}_0 + 2\mathsf{G}_T)$ in FH-CP-ABE scheme. In the same way, when k is fixed, the parameter linearly increases by $2|\mathsf{A}_T|\mathsf{G}_T$ as the j grows in our scheme. We can also find that the decryption cost follows a linear relationship $2\mathsf{G}_T$ with the number of files k when j is fixed in FH-CP-ABE scheme. Accordingly, if the parameter k is fixed, the decryption cost is closed to $|\mathsf{A}_T|\mathsf{G}_T$ with the number of nodes j in our scheme. At the same time, the two parameters in FH-CP-ABE scheme are smaller than CP-ABE's as the j is relatively small in the improved encryption algorithm. Moreover, the length of PK, MSK and SK is equal in the two schemes. We can also observe that the size of CT will linearly increase by $L_{\mathsf{G}_0} + L_{\mathsf{G}_T}$ when an extra document is added and j is a fixed number in FH-CP-ABE scheme. And the value in CP-ABE scheme is larger than FH-CP-ABE's. Similarly, when k is fixed, the parameter also increases linearly as the j grows, and the growth rate is $|\mathsf{A}_T|L_{\mathsf{G}_T}$.

B. Experimental Simulation

To validate theoretical analysis presented in previous subsection, we implement FH-CP-ABE scheme based on the cpabe toolkit and the Java Pairing-Based Cryptography library(JPBC)[42]. The implementation uses a 160-bit elliptic curve group based on the supersingular curve $y^2 = x^3 + x$ over a 512-bit finite field. Meanwhile, to compare experimental results of the encryption and decryption, we also simulate the typical CP-ABE [11] system. In addition, the following experiments are conducted by using Java on the system with Intel Core processor at 2.79 GHz and 1.96GB RAM running Windows XP SP 3. And all of the results are averages of 10trials.

In the simulation, the FH-CP-ABE scheme's implementation adopts the improved encryption algorithm in encryption operation. As in [40], in a CP-ABE scheme, the complexity of access policy associated with ciphertext impacts two aspects. The one is the time cost of encryption and decryption. The other is the storage cost of ciphertext. To illustrated this, we assume that a patient sets his PHR's access policy with k access levels in the form of $\{(a_1, a_2, \ldots, a_i, i \text{ of } i) \text{ AND } a_{i+1} \ldots \text{ AND } a_N\}$(i.e., the worst situation over the policy), where each a_i $(i \in [1, N])$ denotes an attribute. Meanwhile, the patient generates k policies based the above form for using in CP-ABE scheme. For example, assume that the patient shares three files, i.e., $M = \{m_1, m_2, m_3\}$, with three access levels, the access policy is designed as $\{(a_1, a_2, \ldots, a_i, i \text{ of } i) \text{ AND } a_{i+1} \text{ AND } a_{i+2}\}$ in FH-CP-ABE scheme. Accordingly, he should construct three access policies for CP-ABE scheme, where the policies are $\{(a_1,a_2,\ldots,a_i,i \text{ of } i) \text{AND } a_{i+1} \text{ AND } a_{i+2}\}$, $\{(a_1,a_2,\ldots,a_i,i \text{ of } i) \text{ AND } a_{i+1}\}$, and $\{a_1,a_2,\ldots, a_i, i \text{ of } i\}$. The policies only contain AND gate to ensure that all the ciphertext components are computed in decryption algorithm.

The experimental results are given in Fig. 7. Fig. 7(a) shows the time cost of encryption and decryption, while Fig. 7(c) shows the storage cost of ciphertext for various attributes with two hierarchy files. Fig. 7(b) and Fig. 7(d) show the time cost and the storage cost for various files with fixed attributes $N = 30$, respectively. In addition, for various attributes and files, the numbers of attributes and files used in the simulation are N={10,15,20,25,30,35,40,45,50}and $k = \{2, 4, 6, 8\}$.

As illustrated in Fig. 7(a), we can find that the proposed scheme improves the efficiencies of encryption and decryption greatly when two hierarchy files are shared. We can also find

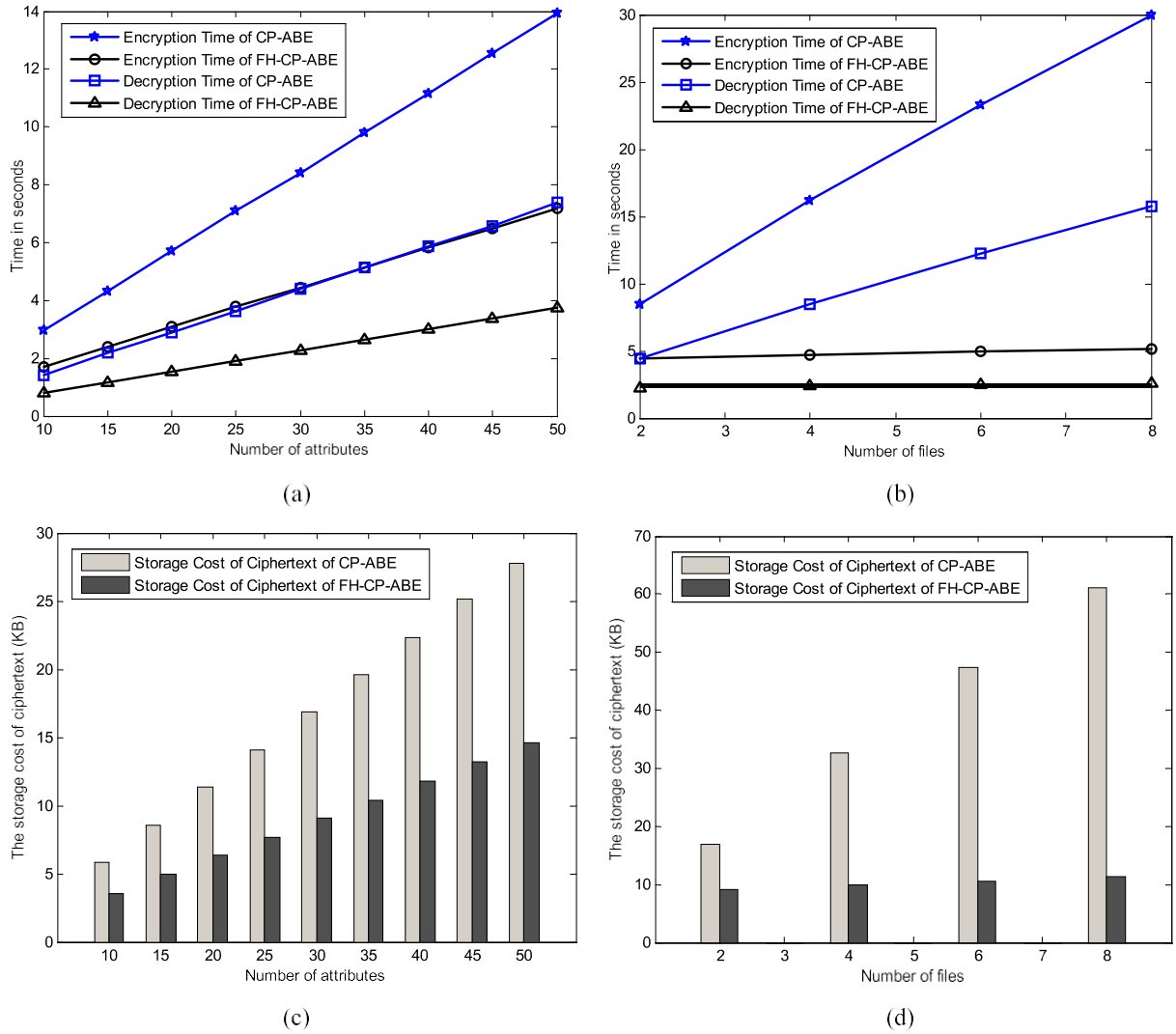


Fig. 7. Comparisons between CP-ABE and FH-CP-ABE. (a) Comparison of the encryption and decryption time cost for two files. (b) Comparison of the encryption and decryption cost under 30 attributes. (c) Storage cost comparison of two ciphertext files. (d) Storage cost comparison of multiple ciphertext files under 30attributes.

that the results are gradually increasing and approximately following a linear relationship with the number of attributes in Fig. 7(a). When the number of files is fixed, the more the number of attributes is used, the more time cost of encryption and decryption in FH-CP-ABE scheme is saved. For example, in Fig. 7(a), the encryption costs of FH-CP-ABE and CP-ABE scheme are 1.8s and 3s approximately when $N = 10$. Similarly, the values are 7s and 14s when $N = 50$. The difference jumps from 1.2s to 7s when N is changed from 10 to 50. For the storage cost of ciphertext, we find that the value in FH-CP-ABE scheme is smaller than CP-ABE's and follows a linear relationship approximately as the number of attributes grows as shown in Fig. 7(c). If the number of files is fixed, the more the number of attributes is used, the higher efficiency in our scheme is improved in terms of storage cost of ciphertext. For example, in Fig. 7(c), when $N = 20$ and $N = 50$, the approximate storage costs of ciphertext are equal to 6.3KB and 14.6KB in FH-CP-ABE scheme, and the values are 11.3KB and 27.8KB in CP-ABE scheme. The saved storage cost is approximately 44.2% and 47.5% with N changed from 20 to 50.

When two hierarchy files are shared, the performance of FH-CP-ABE scheme is better than CP-ABE's in terms of encryption and decryption's time cost, and CT 's storage cost. The reason is described as below. As shown in Fig. 7(a) and Fig. 7(c), the number of files is $k = 2$. Based on the Table I, the encryption time in CP-ABE and FH-CP-ABEschemecan besimplifiedas$[2(|A_{C1}|+|A_{C2}|)+ 2]G_0 + 4G_T$ and $(2|A_{C1}| + 2)G_0 + (2j|A_T| + 4)G_T$, respectively, where j and $|A_T|$ are relatively small in proposed encryption operation. It shows that our scheme can save more encryption time with more common attributes $|A_{C2}|$. Similarly, the storage cost of CT in both schemes can be denoted as $[2(|A_{C1}| + |A_{C2}|) + 2]L_{G_0} + 2L_{G_T}$ and $(2|A_{C1}| + 2)L_{G_0} + (j|A_T| + 2)L_{G_T}$, respectively. It indicates that the FH-CP-ABE scheme has smaller storage cost of ciphertext than the CP-ABE in the same condition. In addition, the decryption time for CP-ABE andFH-CP-ABE

is approximate to $(4|A_u| + 2)C_e + [2(|S_1| + |S_2|) + 4]G_T$ and $(2|A_u|+1)C_e+[2|S_1|+(j|A_T|+4)]G_T$,respectively. Obviously, FH-CP-ABE scheme has lower decryption cost with a same number of common nodes $|S_2|$. So, comparing with CP-ABE and FH-CP-ABE, the encryption cost in our scheme is decreased by $(2|A_{C2}|G_0 - 2j|A_T|G_T)$ in Fig. 7(a). Similarly, the storage cost of CT and the decryption cost in FH-CP-ABE scheme are reduced by $2|A_{C2}|L_{G_0} - j|A_T|L_{G_T})$ and $(2|A_u| + 1)C_e + [2|S_2| - j|A_T|]G_T$ inFig.7(c)andFig.7(a),respectively,comparedtoCP-ABE's. The above simulation results also confirm to theoretical analysis described in previous subsection.

As shown in Fig. 7(b), we find that the results of encryption and decryption cost are approximately following a linear relationshipasthenumberoffilesgrowsinbothschemes.And the values in FH-CP-ABE scheme are smaller than CP-ABE's at the same condition. This figure shows that the time cost of encryption and decryption to be saved is directly proportional to the number of files to be encrypted and decrypted in FH-CP-ABE scheme when the number of attributes is fixed. For example, in Fig. 7(b), the encryption time of CP-ABE scheme is 8.5s and 30s approximately when $k = 2$ and $k = 8$. Accordingly, the parameter in FH-CP-ABE scheme is 4.5s and 5.2s. The difference jumps from 4s to 24.8s with k increased from 2 to 8. In addition, we can find that the experiment result of CT's storage cost in FH-CP-ABE is significantly smaller than CP-ABE scheme's for various files with fixed number of attributes in Fig. 7(d). And this figure indicates that the number of files to be encrypted is proportional to the storage cost of ciphertext to be reduced in the proposed scheme. For example, in Fig. 7(d), when $k = 4$ and $k = 8$, the approximate storage costs of ciphertext are 9.9KB and 11.4KB in FH-CP-ABE scheme, and the values are 32.6KB and 61KB in CP-ABEscheme.Comparingwith CP-ABE and FH-CP-ABE, thereduced storage costisapproximate to 69.6% and 81.3% with k changed from 4to8.Above all, when multiple hierarchy fileswithdifferentaccess levels are shared, the experiment resultsindicatethatFH-CP-ABE scheme performs better than CP-ABEintermsof the time cost of encryption and decryption,andstoragecost of CT , if the number of attributes is fixed. The reason is described as follows. In Fig. 7(b) and Fig. 7(d), the number of attributes is fixed as $N = 30$. Based on the Table I, the encryption cost in CP-ABE and FH-CP-ABE scheme can be denoted as $[2(|A_{C1}| + \ldots + |A_{Ck}|) + k]G_0 + 2kG_T$ and $(2|A_{C1}| + k)G_0 + (2j|A_T| + 2k)G_T$,where$k$={2,4,6,8} in the simulation, and j and $|A_T|$ are relatively small in our proposed scheme. It indicates that FH-CP-ABE requires less encryption time than CP-ABE with more hierarchy files k. Similarly, the CT 's storage cost in both schemes can be denoted as $[2(|A_{C1}| + \ldots + |A_{Ck}|) + k]L_{G_0} + kL_{G_T}$ and $(2|A_{C1}| + k)L_{G_0} + (j|A_T| + k)L_{G_T}$,respectively.Andthe decryptiontimeforCP-ABEandFH-CP-ABEisapproximate to$k(2|A_u|+1)C_e+[2(|S_1|+\ldots+|S_k|)+2k]G_Tand(2|A_u|+1)C_e +[2|S_1| +(j|A_T| +2k)]G_T$,respectively.Theresults show that FH-CP-ABE scheme can save more storage cost and decryption time than CP-ABE under the same condition. In Fig. 7(b), the time cost of encryption and decryption in FH-CP-ABEschemeisreducedby$[2(|A_{C2}|+\ldots+|A_{Ck}|)G_0 - 2j|A_T|G_T]and(k-1)(2|A_u|+1)C_e+[2(|S_2|+\ldots+|S_k|)- j|A_T|]G_T$, respectively, compared to CP-ABE's. Similarly, comparing with CP-ABE and FH-CP-ABE, the storage cost of CT in our scheme is decreased by $[2(|A_{C2}| + \ldots + |A_{Ck}|)L_{G_0} - j|A_T|L_{G_T}]$ in Fig. 7(d). Meanwhile, the results are also consistent with theoretical analysis presented in previous subsection.

V. Conclusions

In this paper, we proposed a variant of CP-ABE to efficiently share the hierarchical files in cloud computing. The hierarchical files are encrypted with an integrated access structure and the ciphertext components related to attributes could be shared by the files. Therefore, both ciphertext storage and time cost of encryption are saved. The proposed scheme has an advantage that users can decrypt all authorization files by computing secret key once. Thus, the time cost of decryption is also saved if the user needs to decrypt multiple files. Moreover, the proposed scheme is proved to be secure under DBDHassumption.

References

[1] C.-K. Chu, W.-T. Zhu, J. Han, J.-K. Liu, J. Xu, and J. Zhou, "Security concerns in popular cloud storage services," *IEEE Pervasive Comput.*, vol. 12, no. 4, pp. 50–57, Oct./Dec. 2013.

[2] T. Jiang, X. Chen, J. Li, D. S. Wong, J. Ma, and J. Liu, "TIMER: Secure andreliable cloudstorageagainstdatare-outsourcing," in*Proc.10thInt. Conf. Inf. Secur. Pract. Exper.*, vol. 8434. May 2014, pp. 346–358.

[3] K. Liang, J. K. Liu, D. S. Wong, and W. Susilo, "An efficient cloud-based revocable identity-based proxy re-encryption scheme for public clouds data sharing," in *Proc. 19th Eur. Symp. Res. Comput. Secur.*, vol. 8712. Sep. 2014, pp. 257–272.

[4] T. H. Yuen, Y. Zhang, S. M. Yiu, and J. K. Liu, "Identity-based encryption with post-challenge auxiliary inputs for secure cloud applications and sensor networks," in *Proc. 19th Eur. Symp. Res. Comput. Secur.*, vol. 8712. Sep. 2014, pp. 130–147.

[5] K. Liang *et al.*, "A DFA-based functional proxy re-encryption scheme for secure public cloud data sharing," *IEEE Trans. Inf. Forensics Security*, vol. 9, no. 10, pp. 1667–1680, Oct. 2014.

[6] T. H. Yuen, J. K. Liu, M. H. Au, X. Huang, W. Susilo, and J. Zhou, "*k*-times attribute-based anonymous access control for cloud comput- ing," *IEEE Trans. Comput.*, vol. 64, no. 9, pp. 2595–2608, Sep. 2015.

[7] J. K. Liu, M. H. Au, X. Huang, R. Lu, and J. Li, "Fine-grained two-factor access control for Web-based cloud computing services," *IEEE Trans.Inf.ForensicsSecurity*,vol.11,no.3,pp.484–497,Mar.2016.

[8] A. Sahai and B. Waters, "Fuzzy identity-based encryption," in *Advances in Cryptology*. Berlin, Germany: Springer, May 2005, pp. 457–473.

[9] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in *Proc. 13th ACM Conf. Comput. Commun. Secur.*, Oct. 2006, pp. 89–98.

[10] W. Zhu, J. Yu, T. Wang, P. Zhang, and W. Xie, "Efficient attribute-based encryption fromR-LWE,"*Chin.J.Electron.*,vol.23,no.4,pp.778–782, Oct.2014.

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

[12] L. Cheung and C. Newport, "Provably secure ciphertext policy ABE," in *Proc. 14th ACM Conf. Comput. Commun. Secur.*, Oct. 2007, pp.456–465.

[13] L. Ibraimi, M. Petkovic, S. Nikova, P. Hartel, and W. Jonker, "Mediated ciphertext-policyattribute-basedencryptionanditsapplication,"in*Proc. 10th Int. Workshop Inf. Secur. Appl.*, Aug. 2009, pp. 309–323.

[14] X. Xie, H. Ma, J. Li, and X. Chen, "An efficient ciphertext-policy attribute-based access control towards revocation in cloud computing," *J. Universal Comput. Sci.*, vol. 19, no. 16, pp. 2349–2367, Oct. 2013.

[15] F. Guo, Y. Mu, W. Susilo, D. S. Wong, and V. Varadharajan, "CP-ABE with constant-size keys for lightweight devices," *IEEE Trans. Inf. Forensics Security*, vol. 9, no. 5, pp. 763–771, May 2014.

[16] A. Balu and K. Kuppusamy, "An expressive and provably secure ciphertext-policy attribute-based encryption," *Inf. Sci.*, vol. 276, pp. 354–362, Aug.2014.

[17] X. Liu, J. Ma, J. Xiong, and G. Liu, "Ciphertext-policy hierarchical attribute-based encryption for fine-grained access control of encryption data," *Int. J. Netw. Secur.*, vol. 16, no. 6, pp. 437–443, Nov. 2014.

[18] Y. Chen, Z. L. Jiang, S. M. Yiu, J. K. Liu, M. H. Au, and X. Wang,"Fully secure ciphertext-policy attribute based encryption with security mediator," in *Proc. 16th Int. Conf. Inf. Commun. Secur.*, vol. 8958. Dec. 2014, pp.274–289.

[19] Y. Yang, J. K. Liu, K. Liang, K.-K. R. Choo, and J. Zhou, "Extended proxy-assistedapproach:Achievingrevocablefine-grainedencryptionof cloud data," in *Proc. 20th Eur. Symp. Res. Comput. Secur. (ESORICS)*, vol. 9327. Sep. 2015, pp. 146–166.

[20] J. Liu, X. Huang, and J. K. Liu, "Secure sharing of personal health records in cloud computing: Ciphertext-policy attribute-based signcryption," *Future Generat. Comput. Syst.*, vol. 52, pp. 67–76, Nov. 2015.

[21] K. Liang *et al.*, "A secure and efficient ciphertext-policy attribute-based proxy re-encryption for cloud data sharing," *Future Generat. Comput. Syst.*, vol. 52, pp. 95–108, Nov. 2015.

[22] C.-I. Fan, V. S.-M. Huang, and H.-M. Ruan, "Arbitrary-state attribute-based encryption with dynamic membership," *IEEE Trans. Comput.*, vol. 63, no. 8, pp. 1951–1961, Aug. 2014.

[23] H. Zheng, Q. Yuan, and J. Chen, "A framework for protecting personal information and privacy," *Secur. Commun. Netw.*, vol. 8, no. 16, pp. 2867–2874, Nov.2015.

[24] F. Xhafa, J. Wang, X. Chen, J. K. Liu, J. Li, and P. Krause, "An efficient PHR service system supporting fuzzy keyword search and fine-grained access control," *Soft Comput.*, vol. 18, no. 9, pp. 1795–1802, Sep. 2014.

[25] S. Wang, J. Yu, P. Zhang, and P. Wang, "A novel file hierarchy access control scheme using attribute-based encryption," *Appl. Mech. Mater.*, vols. 701–702, pp. 911–918, Jan. 2015.

[26] C. Gentry and A. Silverberg, "Hierarchical ID-based cryptography," in *Advances in Cryptology*. Berlin, Germany: Springer, Dec. 2002, pp.548–566.

[27] G. Wang, Q. Liu, and J. Wu, "Hierarchical attribute-based encryption for fine-grained access control in cloud storage services," in *Proc. 17th ACM Conf. Comput. Commun. Secur.*, Oct. 2010, pp. 735–737.

[28] 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, no. 2, pp. 743–754, Apr.2012.

[29] X. Zou, "A hierarchical attribute-based encryption scheme," *Wuhan Univ. J. Natural Sci.*, vol. 18, no. 3, pp. 259–264, Jun. 2013.

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

[31] B. Waters, "Ciphertext-policy attribute-based encryption: An expressive, efficient,andprovablysecurerealization,"in*Proc.14thInt.Conf.Pract. Theory Public Key Cryptogr. (PKC)*, vol. 6571. Mar. 2011, pp. 53–70.

[32] T. Nishide, K. Yoneyama, and K. Ohta, "Attribute-based encryption with partially hidden encryptor-specified access structures," in *Proc. 6th Int. Conf. Appl. Cryptogr. Netw. Secur.*, vol. 5037. Jun. 2008, pp. 111–129.

[33] K. Emura, A. Miyaji, A. Nomura, K. Omote, and M. Soshi, "A ciphertext-policy attribute-based encryption scheme with constant ciphertext length," in *Proc. 5th Int. Conf. Inf. Secur. Pract. Exper.*, vol. 5451. Apr. 2009, pp. 13–23.

[34] V. Goyal, A. Jain, O. Pandey, and A. Sahai, "Bounded ciphertext policy attribute based encryption," in *Proc. 35th Int. Colloq. Automata, Lang. Program.*, vol. 5126. Jul. 2008, pp. 579–591.

[35] X. Liang, Z. Cao, H. Lin, and D. Xing, "Provably secure and efficient bounded ciphertext policy attribute based encryption," in *Proc. 4th Int. Symp. Inf., Comput., Commun. Secur.*, Mar. 2009, pp. 343–352.

[36] L. Ibraimi, Q. Tang, P. Hartel, and W. Jonker, "Efficient and provable secure ciphertext-policy attribute-based encryption schemes," in *Proc. 5th Int. Conf. Inf. Secur. Pract. Exper.*, vol. 5451. Apr. 2009, pp. 1–12.

[37] A. Lewko, T. Okamoto, A. Sahai, K. Takashima, and B. Waters, "Fully secure functional encryption: Attribute-based encryption and (hierarchical) inner product encryption," in *Proc. 29th Annu. Int. Conf. Theory Appl. Cryptogr. Techn.*, vol. 6110. May 2010, pp. 62–91.

[38] J. Hur, "Improving security and efficiency in attribute-based data sharing," *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 10, pp. 2271–2282, Oct. 2013.

[39] M. Green, S. Hohenberger, and B. Waters, "Outsourcing the decryption of ABE ciphertexts," in *Proc. 20th USENIX Conf. Secur.*, Aug. 2011, pp.1–16.

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

[41] S. Hohenberger and B. Waters, "Online/offline attribute-based encryption,"in*Proc.17thInt.Conf.Pract.TheoryPublic-Key Cryptogr.(PKC)*, vol. 8383. Mar. 2014, pp. 293–310.

[42] A. De Caro and V. Iovino, "jPBC: Java pairing based cryptography," in *Proc. IEEE Symp. Comput. Commun.*, Jun./Jul. 2011, pp. 850–855.