An enhanced media ciphertext-policy attribute-based encryption algorithm on media cloud

Hao Li\textsuperscript{1,2}, Lianbing Deng\textsuperscript{1}, Cheng Yang\textsuperscript{3} and Jianbo Liu\textsuperscript{3}

Abstract
With the development of cloud computing, more and more video services are moving to the cloud. How to realize fine-grained access control of those data on cloud becomes an urgent problem. Attribute-based encryption provides a solution. However, heavy computation is still a bottleneck restricting the wider application of attribute-based encryption in cloud computing. In addition, we find that expression of the access control structure on media cloud can be further improved. To solve these problems, we propose an enhanced media ciphertext-policy attribute-based encryption algorithm and introduce its two key components, the multiple access tree and the key chain. To increase the scalability of the proposed algorithm, we discussed the issues of multi-authorization, user revocation, and \(1-n\) multiple access tree. Security analysis shows that enhanced media ciphertext-policy attribute-based encryption can successfully resist chosen-plaintext attacks under the decisional bilinear Diffie–Hellman assumption. Performance analysis proves both theoretically and practically that the proposed algorithm incurs less computational cost than the traditional ciphertext-policy attribute-based encryption, multi-message ciphertext-policy attribute-based encryption, and scalable ciphertext-policy attribute-based encryption by optimizing the access control structure. Our proposed algorithm has strong practical significance in media cloud.

Keywords
Access control, ciphertext-policy attribute-based encryption, video encryption, fine-grained, access structure, key chain

Introduction
Media cloud provides video storage, processing, and distribution services to users. How to realize fine-grained access control of massive amounts of videos is a huge challenge for the media cloud.\textsuperscript{1} Ciphertext-policy attribute-based encryption (CP-ABE) is an important technology to tackle this challenge in application scenarios such as personal health records (PHR)\textsuperscript{2,3} and smart grids.\textsuperscript{4,5}

In CP-ABE, conversion from the access control policy to the access structure is an important step. However, when CP-ABE is applied in media cloud, the traditional conversion methods become inefficient. On media cloud, one video file is often available with different resolutions. The fact is that users with access to high-resolution videos are always allowed to access low-resolution versions. For example, paying users can watch a movie in all three versions: the standard definition (SD) version, the high definition (HD) version, the

---

\textsuperscript{1}Da Hengqin Science and Technology Development Co., Ltd., Zhuhai, China
\textsuperscript{2}Communication University of China, Beijing, China
\textsuperscript{3}School of Information and Communication Engineering, Communication University of China, Beijing, China

Corresponding author:
Cheng Yang, School of Information and Communication Engineering, Communication University of China, No. 1 DingFuZhuang East St., Beijing 100024, China.
Email: chy@cuc.edu.cn
trees. Wu et al.\textsuperscript{7} presented a multi-message CP-ABE algorithm. Then, eM-CP-ABE is applied to media cloud for fine-grained access control. Two steps are required: first, the video is divided into segments according to resolution or time and encrypted by the advanced encryption standard (AES). Note that different segments correspond to different keys. Second, these symmetric keys are encrypted by eM-CP-ABE to achieve fine-grained access control. The contribution of our proposed algorithm is reflected in the following four aspects:

1. In the encryption process, eM-CP-ABE consumes less time and space than MCP-ABE; in the decryption process, when an authorized user switch from a high-level resource to a low-level resource, only two hash operations need to be performed without any new decryption operations. Compared with SCP-ABE, eM-CP-ABE has better performance both in encryption and decryption processes.

2. Using one $1-n$ MAT, eM-CP-ABE can encrypt several videos with different privilege levels. It significantly reduces the cost of generating and maintaining access structures.

3. eM-CP-ABE has strong scalability. We have proved that both multi-authorization and user revocation are feasible when the eM-CP-ABE algorithm is used.

4. We have also proved security of the eM-CP-ABE algorithm. It can successfully resist chosen-plaintext attacks (CPAs) under the decisional bilinear Diffie–Hellman (DBDH) assumption.

The rest of the article is organized as follows: section “Related work” provides an overview of the related work. Section “Preliminary” presents necessary preliminary knowledge. Section “Proposed algorithm” introduces eM-CP-ABE and its extensions. Section “Security analysis” analyzes eM-CP-ABE in terms of security. Section “Performance analysis” analyzes the operation overheads of the proposed algorithm. Finally, section “Conclusion” presents the conclusion.

### Related work

CP-ABE was first proposed in 2007.\textsuperscript{9} In CP-ABE, user keys are associated with sets of attributes, whereas ciphertexts are associated with policies. CP-ABE is widely used in fine-grained access control of data outsourced on the cloud. However, the problem is that CP-ABE entails heavy computation and complex conversion from access policy to access structure. Heavy computation will degrade real-time performance and increase the workload of computation. To optimize real-time performance, Li et al.\textsuperscript{10} introduced online/offline ABE. A similar online/offline scheme was proposed by Cui et al.\textsuperscript{11} The online/offline transfer technology allows the computation tasks to be outsourced to public cloud servers. However, the online/offline transfer technology can just transfer the workload of computation, but cannot reduce it. Expanding the single authorization center to multi-authorization

### Table 1. Access control policies in media cloud.

| $r$  | (user = Guest), (vid = 10010, category = story, quality = SD), play) |
|------|--------------------------------------------------------------------|
| $r_2$ | (user = Registered), (vid = 10010, category = story, quality = (SD, HD)), play) |
| $r_3$ | (user = Paying), (vid = 10010, category = story, quality = (SD, HD, UHD)), play) |

SD: standard definition; HD: high definition; UHD: ultra-high definition.
centers can improve the efficiency of key generation. Han et al.\textsuperscript{12} introduced a decentralized CP-ABE scheme, in which each authority can work on one unique attribute subset independently. The robust and auditable access control (RAAC) scheme was proposed in 2017. In RAAC, any authority can manage the whole attribute set individually and an auditing mechanism is introduced to coordinate all authorities’ work.\textsuperscript{13} Multi-authorization greatly improves the performance of key generation, but does not reduce the workload of encryption and decryption in CP-ABE. Xue et al.\textsuperscript{14} proposed an attribute-based controlled access control scheme, in which multiple users having different attribute sets could collaborate to gain access permission. The scheme is efficient in terms of storage and computation overhead, but user collusion is forbidden in most scenarios of the media cloud.

However, addition of many practical factors makes the access structure more expressive. The factor of time was introduced to realize time-release encryption.\textsuperscript{15} In some existing studies,\textsuperscript{16–19} user revocation was supported in their CP-ABE schemes. In studies by Ning et al.,\textsuperscript{20,21} illegitimate access could be traced. However, the problem of complex conversion from access policy to access structure remains unsolved. Traditional access trees are inefficient when the corresponding access control policies have the inclusion relationships such as file system. In hierarchical file systems, if a user can decrypt a file corresponding to a level node in the access structure tree, he or she can decrypt all these files corresponding to the lower level nodes. To avoid repetitive description of attributes on different levels, Wang et al.\textsuperscript{22} used an integrated access structure to encrypt multiple files on different levels separately. As only one file can be encrypted on each level, the scheme proposed by Yeh et al.\textsuperscript{2} is not practical. Li et al.\textsuperscript{23} proposed an extended file hierarchy CP-ABE scheme (EFH-CP-ABE), which could encrypt several files on the same access level. To simplify the complex structure of inclusion relation among these privilege levels, Bobba et al.\textsuperscript{24} introduced a ciphertext-policy attribute-set-based encryption (CP-ASBE) scheme in which the set of user attributes was extended into a recursive set structure. CP-ASBE was used in the conditional access system to resist collusion attacks.\textsuperscript{25} Based on CP-ASBE, Wan et al.\textsuperscript{26} proposed a hierarchical attribute-set-based encryption (HASBE) algorithm. HASBE builds a hierarchical system model and realizes the key delegation scheme and user revocation scheme. Teng et al.\textsuperscript{27} improved the HASBE algorithm by keeping the length of the ciphertext and the number of bilinearity operation fixed. However, if strict mutual exclusion cannot be satisfied between different privilege levels, the common attributes will still be described repeatedly.

The same problem is faced in media cloud. CP-ABE is widely used to ensure the security of outsourced videos and support flexible video services.\textsuperscript{7,8,28} Yang et al.\textsuperscript{28} used traditional access trees that could not improve the efficiency. Access trees both in MCP-ABE\textsuperscript{7} and SCP-ABE\textsuperscript{8} can represent the access control rules with full inclusion relation efficiently. However, when there is any common attribute, these trees become less efficient because the common attributes have to be described repeatedly on different levels of the tree.

In this article, we propose a new access tree called MAT and a new one-way key chain scheme. On the basis of these two components, we introduce a novel encryption algorithm called eM-CP-ABE. In the encryption process, one $1 - n$ MAT can be flexibly used to encrypt several videos with multiple levels of privilege, thus considerably reducing the cost of generating and maintaining access structures. In the decryption process, user on the $j$th level will get two secret numbers, $a_j$ and $a_0$, and the key corresponding to this level is $k_j = H_2(a_j||a_0)$. When the user accesses the low level $i = j - 1$, the key corresponding to the $i$th level is $k_i = H_2(H_2(a_j||b)||a_0)$. Only two hash operations are needed to complete the access control.

**Preliminary**

To ensure the independence of the article, the relevant information is given in this section.

**Bilinear maps**

Let $\mathbb{G}_0, \mathbb{G}_1$ be two multiplicative cyclic groups of prime order $p$. Let $g$ be a generator of $\mathbb{G}_0$ and $e$ be a bilinear map, $e : \mathbb{G}_0 \times \mathbb{G}_0 \to \mathbb{G}_1$. The bilinear map $e$ has the following properties:

1. **Bilinearity**: $\forall a, b \in \mathbb{Z}_p$ and $g_1, g_1 \in \mathbb{G}_0$, and we have $e(g_1^a, g_1^b) = e(g_1, g_1)^{ab}$.
2. **Non-degeneracy**: There exists $g_1, g_1 \in \mathbb{G}_0$ such that $e(g_1, g_1) \neq 1$, which means the map does not send all pairs in $\mathbb{G}_0 \times \mathbb{G}_0$ to the identity in $\mathbb{G}_1$.
3. **Computability**: There is an efficient algorithm to compute $e(g_1, g_1)$ for all $g_1, g_1 \in \mathbb{G}_0$.

**Access structure**

Let $\{P_1, P_2, \ldots, P_n\}$ be a set of parties. A collection $A \subseteq 2^{\{P_1, P_2, \ldots, P_n\}}$ is monotone if $\forall B, C$; if $B \in A$ and $B \subseteq C$ then $C \in A$. An access structure is a collection $A$ of non-empty subset of $\{P_1, P_2, \ldots, P_n\}$, that is, $A \subseteq 2^{\{P_1, P_2, \ldots, P_n\}} \setminus \{\emptyset\}$. The sets in $A$ are called authorized sets, and the sets not in $A$ are called unauthorized sets.

In CP-ABE, a set of attributes is used as the decryption key, and an access structure is used to encrypt the plaintext. We describe the access structure in terms of
an access tree. If and only if a set of attributes satisfies the access tree, the ciphertext can be decrypted correctly.

**Access tree** $T$

Let $T$ be a tree representing an access structure. A non-leaf node $x$ of the tree is described by its child number $num_x$ and a threshold value $k_x$, where $0 \leq k_x \leq num_x$. We denote the parent of the node $x$ by parent($x$). The children of $x$ are numbered from 1 to $num_x$. The function index ($x$) returns such a number associated with the node $x$. When $x$ is a leaf node, we denote the attribute in $x$ by $\text{attr}(x)$.

**DBDH assumption**

Let $a, b, c, z \in \mathbb{Z}_p$ be chosen at random and $g$ be a generator of $\mathbb{G}_0$. The DBDH assumption is that no probabilistic polynomial-time algorithm $B$ can distinguish the tuple ($A = g^a, B = g^b, C = g^c, e(g, g)_{abc}$) from the tuple ($A = g^a, B = g^b, C = g^c, e(g, g)^2$) with more than a negligible advantage. The advantage of $B$ is

$$\Pr[B(A, B, C, e(g, g)^{abc}) = 0] - \Pr[B(A, B, C, e(g, g)^2) = 0]$$

where the probability is taken over the random choice of the generator $g$, the random choice of $a, b, c, z \in \mathbb{Z}_p$, and the random bits consumed by $B$.

**Proposed algorithm**

In this section, eM-CP-ABE is described in detail. First, two important components that construct the proposed algorithm is introduced. Then, the model and implementation of eM-CP-ABE are proposed. Last, multi-authorization, user revocation, and $1-n$ MAT are discussed.

**Components**

**MAT.** MAT is introduced to improve the expression efficiency of access control tree in eM-CP-ABE. In our article, MAT’s nodes are divided into three types: virtual nodes, attribute nodes, and ordinary nodes. A virtual node $n_v$ is represented by $k_v = 0$ and $num_{n_v} = 1$. A subtree with a virtual node as its parent consists of one “trunk” and some “branches.” Trunk is used to describe the inclusion relation between adjacent levels. Branches are used to describe the particular rules of each level. The parent of a virtual node is represented as $y = \text{parent}(n_v)$. If a parent node $y$ has $num_y$ children nodes, including $num'_y$ virtual nodes, then the threshold value of $y$ satisfies $0 \leq k_y \leq num_y - num'_y$. All leaf nodes are called attribute nodes. The rest of the nodes are called ordinary nodes.

We also make the following assumptions:

1. $num_y > num'_y$: The inequality means that common attributes must exist.
2. The parent of a virtual node cannot be a virtual node.
3. The virtual node has one and only one child ($num_{n_v} = 1$), which consists of one “trunk” and some “branches.”

We define a MAT with $n$ virtual nodes as $1-n$ MAT. Figure 1 shows a $1-2$ MAT.

**Satisfying an access tree.** To clearly describe the process, MATs $T$ in Figure 1 are illustrated as an example. The process can be extended to more complex trees.

**Step 1.** List all hierarchical trees rooted at the virtual node. There are two qualified hierarchical trees ($T_1, T_2$) in Figure 1.

**Step 2.** Delete all virtual nodes and their descendants, and obtain the base tree $T_0$.

**Step 3.** A set of attributes $\gamma$ satisfies $T$, when $T(\gamma) = T_1(\gamma)$ and $T_0(\gamma) = 1$ or $T(\gamma) = T_2(\gamma)$ and $T_0(\gamma) = 1$.

**Key chain.** Without loss of generality, we assume there is a $1-1$ MAT (Figure 2), which is used to encrypt one video with $l$ privilege levels and $m$ common attributes. We assume the video is divided into $l$ units and its first $j$ units are organized into an access level $L_j$. The corresponding user has privilege $p_j, j = 0, 1, \ldots, l$. By following the same operations in the study by Wu et al., we define a binary dominant relation $\preceq$ on the set of privilege $\{p_1, \ldots, p_l\}$: $p_1 \preceq p_2 \preceq \ldots \preceq p_l$.

The goal we need to achieve is that if the user can access the content of level $j$, then the user should at least have the key parameters $a_0, a_j$. Through $a_0, a_j$, the user can deduce $\{a_i : 1 \leq i \leq j\}$. Our design is as follows:

**Step 1.** Find a one-way hash function $H_2(\ast)$, with the following characteristics:

1. Input data of arbitrary lengths and output data of fixed lengths.
2. For a given input, it is easy to calculate the output.
3. For a given output, it is not computationally feasible to find a correct input.
4. $H_2 : \{0, 1\}^* \rightarrow \mathbb{Z}_p^*$. 
Step 2. For the highest privilege level \( l \), we choose a number \( a_l \) uniformly at random from \( \mathbb{Z}_p^\ast \). Then, we choose \( a_0 \) uniformly at random from \( \mathbb{Z}_p^\ast \). The key corresponding to the level \( l \) is \( k_l = H_2(a_l||a_0) \). Then, the number corresponding to the \( j \)th level is \( a_j = H_2(k_j + 1||j) \), and the key corresponding to the level \( j \) is \( k_j = H_2(a_j||a_0) \), where \( 1 \leq j < l - 1 \).

Step 3. The final key chain \( \{k_1, \ldots, k_l\} \) can be used as the keys of a symmetric encryption algorithm, such as AES, to encrypt \( l \) video units separately.

When a set of attributes satisfies both the base tree constructed by the common attributes \( (a_0) \) and the \( j \)th level of the hierarchical trees \( (a_j) \), \( k_j \) can be calculated by \( k_j = H_2(a_j||a_0) \). Also, using \( a_0, a_j, k_j \) can be calculated when \( 1 \leq i < j \). Due to the unidirectionality of \( H_2(\ast) \), the low-level user cannot be calculated to obtain the high-level key.

Using MAT and key chain. In this part, we show how to use MAT and the key chain to convert complex policies into access trees. According to Table 1, we add the time factor to construct more complex access rules (Table 2).

In this scenario, the common attributes (id, category, and duration) of the three rules must be described in each branch of CP-ABE or MCP-ABE, which considerably reduces the expression efficiency of the access control structure.

![Figure 1. 1 – 2 multiple access tree.](image)

**Table 2.** Three access control rules for the video with ID=10010.

|   |   |   |   |   |
|---|---|---|---|---|
| r1 | (user = Guest), <br/> (vid = 10010, category = story, <br/> quality = standard), (duration < 30), play) |
| r2 | (user = Registered), <br/> (vid = 10010, category = story, <br/> quality = (standard, high)), (duration < 30), play) |
| r3 | (user = Paying), (vid = 10010, category = story, <br/> quality = (standard, high, ultra)), (duration < 30), play) |

r1: Guests can only watch the SD-version videos with ID 10010 in the story category this month. (For the sake of simplicity, we consider that there are 30 days in a month.)

r2: Registered users can watch HD- and SD-version videos with ID 10010 in the story category this month.

r3: Paying users can watch UHD-, HD- and SD-version videos with ID 10010 in the story category this month.

The rules that conform to the ABAC format are described in Table 2.

We assume that two bits are used to describe the user’s type (Paying: uType = 11, Registered: uType = 10).
In CP-ABE, three rules will be converted into three access trees (Figure 3). We use the integer comparison technology to describe the expiration time “duration < 30.” $T_1$ is the access tree of $r_1$ and is used to encrypt the key corresponding to the SD video. As shown in Figure 3(a), any user can access the specific SD video. $T_2$ is the access tree of $r_2$ and is used to encrypt the key corresponding to the HD video. As shown in Figure 3(b), only the user whose “uType” attribute value is “11” or “10” has access to the specific HD video. Similarly, in $T_3$, the user with “uType” value of “11” can access the specific UHD video.

In MCP-ABE, the three rules will be converted into one complex access tree (Figure 4). In MCP-ABE, a special access tree is consisted of one “trunk” and some “branches.” As shown in Figure 4, the trunk represents the inclusion relationship between the three rules, $r_1 \subset r_2 \subset r_3$. On each level, the branches represent the details of the corresponding rule. However, the common attributes need to be described on each level repeatedly in MCP-ABE.

We use MAT and key chain to simplify the access tree. As shown in Figure 5, subtrees of the common attributes are extracted to form the base tree, which is then merged with the hierarchical tree through an
Then, for each of the remaining levels, there exist $Z$ in with a simpler tree structure and fewer nodes. CP-ABE algorithm realizes expression of the three rules $\text{Setup}$, $\text{Encrypt}$, $\text{KeyGen}$, $\text{Decrypt}$, and $\text{Delegate}$. Authority center, and trusted encryption center. Media cloud, video provider, video consumer, attribute $\text{ABE}$. The system consists of five participants. They are $\text{eM-CP-ABE}$ model. Figure 6 shows the model of $\text{eM-CP-ABE}$. Compared with CP-ABE and MCP-ABE, the $\text{eM-CP-ABE}$ model applies his or her secret key $SK$. Inputs are $PK$, $CT$, and $SK$ and its output is plaintext $M$.

\text{eM-CP-ABE} algorithm. We take 1–1 MAT (Figure 2) as an example to describe the $\text{eM-CP-ABE}$ algorithm. We assume that one video has $l$ units corresponding to $l$ privilege levels. There is at least one common attribute on the $l$ privilege levels.

\text{Setup}(\lambda). Let $G_0$, $G_T$ be two multiplicative cyclic groups of prime order $p$. Let $g$ be a generator of $G_0$ and $e$ be a bilinear map, and $e : G_0 \times G_0 \rightarrow G_T$. Choose $\alpha$ and $\beta$ uniformly at random from $\mathbb{Z}_p^\ast$, $\alpha, \beta \in \mathbb{Z}_p^\ast$. Then, the public key is published as $PK = (G_0, g, g_1, g_2, f_1)$, where $g_1 = g^\alpha$, $g_2 = e(g, g)^\beta$, $f_1 = g^{\beta}$, and the master key $MK$ is $(\beta, g^\beta)$. $\text{Encrypt}(PK, M, T)$. The $1–1$ MAT $T$ is divided into a base tree $T_0$ and a hierarchical tree $T_1$ (Figure 7). For each node $N$ in trees $T_0$ and $T_1$, we choose a polynomial $f_N$ randomly, set the degree $q_N = k_N - 1$, where $k_N$ is the threshold of node $N$ and set $f_N(0) = f_{\text{parent}(N)}(\text{index}(N))$.

We choose $a_0, a_1, s_0, s_1 \in \mathbb{Z}_p$ uniformly at random from $\mathbb{Z}_p$. $N_R$ denotes the root node of $T_0$ and $N$, denotes the root node of $T_1$. Then, we set $f_{N_0}(0) = s_0$ and $f_{N_1}(0) = s_1$. Using the established polynomial of each

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{figure4.png}
\caption{Access tree corresponding to the three rules in MCP-ABE.}
\end{figure}
node, the secrets, $s_0, s_1$, are shared. The set of leaf nodes in $T$ is denoted by $N_{\text{leaf}}$, where $n_i \in N_{\text{leaf}}$. The set of nodes in $T_1$’s trunk is denoted by $\{N_1, \ldots, N_l\}$.

The ciphertext has two parts. In the first part, a symmetric encryption algorithm $E(m, k)$ is used to encrypt each video unit. The second part is ABE. The whole ciphertext is expressed as

$$CT_s = EM||CT$$

$$EM = \{E(m, k)\}_{i=1}^l, \quad \text{where } k_i = H_2(a_i|a_0)$$

$$CT = \left(\begin{array}{c}
B_0, C_0, B_1, C_1, \{B_{N_i}, C_{N_i}\}_{i=2}^l, \{E_{n_i}, E_n^i\}_{n_i \in N_{\text{leaf}}} \end{array}\right)$$
where
\[ B_0 = a_0e(g, g)^{\alpha_0}, C_0 = g^b_{0_0}, B_1 = a_1e(g, g)^{\alpha_1}, C_1 = g^b_{1_0} \]
\[ B_{N_i} = a_ie(g, g)^{\alpha_i}, a_i = H_2(k_i + ||j||_0), C_{N_i} = g^b_{N_i} \]
\[ E_{n_i} = g^{\alpha_i}, E'_{n_i} = H_1(\text{att}(n_i))^s_i \]

\[ H_1(\cdot) \] is a collision-resistant hash function: 3
\[ H_1 : \{0, 1\}^* \rightarrow \mathbb{Z}^*_p. \] This means it is hard to find two pre-values that can generate the same hash value.

KeyGen(MK, S). We choose \( r \in \mathbb{Z}^*_p \) uniformly at random from \( \mathbb{Z}^*_p \). Then, we randomly choose \( r_j \in \mathbb{Z}^*_p, \forall j \in S \). The secret key \( SK \) is
\[ SK = \{D_j, \{D'_j, D''_j\} \}_{j \in S} \]
where \( D = g^{(\alpha + \beta)/2}, D_j = g^r H_1(j)^{g/2} \) and \( D'_j = g^{r_j} \).

Decrypt(PK, CTs, SK). The video consumer does the following operations to realize decryption.

Step 1. To determine whether the set of attributes \( S \) satisfies the access tree \( T : T(S) = T_0(S) \) and \( T_1(S) \). When \( T(S) = 1 \), go to Step 2; otherwise, return.

Step 2. We define DecryptNode(CTs, SK, x) as a recursive algorithm, where \( x \) is a node in \( T_0 \). The following two cases are discussed.

Case 1. \( x \) is a leaf node, if \( i = \text{att}(x) \in S \)
\[ \text{DecryptNode}(CTs, SK, x) = \frac{e(D_i, E_i)}{e(D_i', E_i')} = \frac{e(g^{r_i} H_1(i)^{g/2})}{e(g^{r_i}, H_1(\text{att}(x))^{g/2})} = e(g, g)^{r_i} \]
else \( i = \text{att}(x) \notin S \), DecryptNode(CTs, SK, x) = ⊥.

Case 2. \( x \) is a non-leaf node. Considering that all node \( Z = \{z_1, z_2, \ldots, z_n\} \) are children of \( x \), we set \( F_x = \{F_{z_i} = \text{DecryptNode}(CTs, SK, z_i) \mid z_i \in Z \}. \) Let \( S_i \) be an arbitrary \( k_i \)-sized set of child nodes \( z_i \) such that \( F_{z_i} \neq \perp \). If no such set exists, then the node is not satisfied and the function returns ⊥.

Otherwise, we compute
\[ F_x = \prod_{z \in S_i} F^\Delta_{i, s_i}(0) \]
\[ = \prod_{z \in S_i} (e(g, g)^{r_i})^\Delta_{i, s_i}(0) \]
\[ = \prod_{z \in S_i} (e(g, g)^{r_i}_{\text{max}(|\text{index}(z)|) + |\text{index}(z)|})^\Delta_{i, s_i}(0) \]
\[ = \prod_{z \in S_i} (e(g, g)^{r_i})^\Delta_{i, s_i}(0) \]
\[ = e(g, g)^{r_i} \]

When \( x = N_R \), we obtain \( A = \text{DecryptNode}(CTs, SK, N_R) = e(g, g)^{r_0} \) after recursion. Then
\[ \frac{B_0}{(e(C_0, D)/A)} = \frac{B_0}{e(C_0, D)} = \frac{(e(g^{b_{0_0}}, g^{\alpha + \beta/2})/e(g, g)^{\alpha_0})}{e(g, g)^{\alpha_0}} = a_i \]

Similarly, if \( x = N_i \in \{N_1, \ldots, N_i\} \) in \( T_1 \), we obtain \( A = \text{DecryptNode}(CTs, SK, N_i) = e(g, g)^{r_i} \).

Then
\[ \frac{B_{N_i}}{(e(C_{N_i}, D)/A)} = \frac{B_{N_i}}{e(C_{N_i}, D)} = \frac{e(g^{b_{N_i}}, g^{\alpha + \beta/2})/e(g, g)^{\alpha_{N_i}}}{e(g, g)^{\alpha_{N_i}}} = a_i \]

Step 3. The key of the \( i \)th level \( k_i = H_2(k_i || a_0) \) is obtained. Using the key chain, the keys used by lower levels are generated.

Extension

Multi-authorization. In the cloud environment, when a large number of users access a hot video content in a short time, it poses a big challenge for the attribute authorization center. Extending the single authorization center to multiple centers can alleviate the pressure effectively. Using KeyGen and Delegate functions, the single authorization center is extended to authorization centers (Figure 8).

We assume that each attribute authorization center can complete the authorization service independently: AA \( A \rightarrow S_q \) and \( |S_q| \geq |S_{\text{max}}| \). For the attribute authorization center, there exists AA \( \text{Center} \rightarrow S \).

Multi-authorization is realized as follows.

![Figure 8. Hierarchical authorization group.](image-url)
Case 1. Apply for a secondary $AA_{i(n+1)} \rightarrow S_{i(n+1)}$ from $AA_{Center} \rightarrow S$.

$AA_{Center}$ has $n$ children already, called $AA_{i1}, \ldots, AA_{in}$, and the corresponding attribute sets are $S_{i1}, \ldots, S_{in}$. $AA_{i(n+1)} \rightarrow S_{i(n+1)}$, satisfying $|S_{i(n+1)}| \geq |S_{Max}|$ and $S_{i1} \cup \ldots \cup S_{in} \cup S_{i(n+1)} \subseteq S$ wants to join the secondary AA.

**Step 1.** Check the legitimacy of $AA_{i(n+1)}$.
**Step 2.** Choose $r \in \mathbb{Z}_p^*$ and $t_j \in \mathbb{Z}_p$, $\forall j \in S_{i(n+1)}$.

$$MK_{i(n+1)} = \text{KeyGen}(MK, S_{i(n+1)})$$

**Step 3.**

$$\left( S_{i(n+1)}, D_j \left\{ D_{j}, D_{j}' \right\} \right)_{j \in S_{i(n+1)}}$$

where $D = g^{(\alpha + r)/\beta}$, $D_j = g^\gamma H_1(f_j)$, $D_{j}' = g^r$.

**Step 4.** $AA_{Center}$ sends $MK_{i(n+1)}$ to $AA_{1(n+1)}$.

**Case 2.** Apply for an ordinary $AA_{i(n+1)} \rightarrow S_{i(n+1)}$ from $AA_{i-1} \neq AA_{Center} \rightarrow S$.

**Step 1.** Check the legitimacy of $AA_{i(n+1)}$.

**Step 2.** Choose $r \in \mathbb{Z}_p^*$ and $t_k \in \mathbb{Z}_p$, $\forall k \in S_{i(n+1)}$.

$$MK_{i(n+1)} = \text{Delegate}(MK_{i-1}, S_{i(n+1)})$$

**Step 3.**

$$\left( S_{i(n+1)}, \tilde{D}_k \left\{ \tilde{D}_k, \tilde{D}_k' \right\} \right)_{k \in S_{i(n+1)}}$$

where $\tilde{D} = Df_k^{\gamma}$, $\tilde{D}_k = D_k g^\gamma H_1(k)^\gamma$ and $\tilde{D}_k' = D_k' g^{\gamma}$.

**Step 4.** $AA_{i-1}$ sends $MK_{i(n+1)}$ to $AA_{i(n+1)}$.

**User revocation.** To realize user revocation, eM-CP-ABE adds timestamps to the MAT and attribute sets separately. Compared with CP-ABE, eM-CP-ABE is more flexible when the timestamp is added in a branch, then the algorithm implements user revocation on the corresponding privilege level. When the timestamp is a set of common attributes, the algorithm implements user revocation on all privilege levels, that is, the plaintext $M$ is encrypted under the timestamp $x$ and the expiration time $y$ labeled in a user’s attribute set. When $y \leq x$, the user cannot decrypt the ciphertext correctly.

$1-n$ MAT. When using ABE, different access control trees need to be established for data with different access requirements. Generation and maintenance of a large number of access trees is a tough task. A $1-n$ MAT can describe the access structure of $n$ videos with different access rules. In $1-n$ MAT, the base tree is designed to describe the common rules of the videos, and $n$ hierarchical trees are used to describe the private rules of different videos. This method reduces both the number of trees and the operations required for encryption.

Figure 9 shows a $1-n$ MAT. The base tree has $m_0$ common attributes and $n$ hierarchical trees. We assume that the $j$th video has $t_j (1 \leq j \leq n)$ privilege levels, and the ciphertext can be obtained as follows

$$CT_s = \left( CT_j^s \right)_{j=1}^n$$

where $CT_j^s$ is the $j$th encrypted video.
CT' = \{ EM_j || CT_j, EM_j = \{ E(m_{j,i}, h_{j,i}) \}_{i=1}^k \}

CT_j = \{ B_{j,0}, C_{j,0}, B_{j,1}, C_{j,1}, \{ B_{N_{j,i}, C_{N_{j,i}}} \}_{i=2}^n, \{ E_{a,n}, E'_{a,n} \}_{a \in \mathbb{T}_{\text{out}}}, T \}

where \( T \) is the \( 1 - n \) MAT.

**Security analysis**

**Selective-set model**

There are four common types of attacks: ciphertext only, known plaintext, chosen plaintext, and chosen ciphertext. The CPA is the most powerful type of attack. If an algorithm can resist this attack, it can also resist other types of attack. To prove that eM-CP-ABE is secure, we first define a selective-set model based on CPA. The model is described as follows:

**Init.** The adversary declares the \( 1 - n \) MAT, \( T^* \), that he wants to be challenged upon. The \( T^* \) is sent to the challenger.

**Setup.** The challenger runs the Setup algorithm, and then gives the public parameters to the adversary.

**Phase 1.** The adversary is allowed to query for private keys for many attribute sets \( S_1, \ldots, S_q \), where \( T^*(S_i) = 0 \) for all \( i \in [1, q] \).

**Challenge.** The adversary submits two messages of equal length, \( M_0 \) and \( M_1 \). The challenger flips a random coin \( \mu \), and encrypts \( M_{\mu} \) with \( T^* \). The ciphertext \( CT^* \) is passed to the adversary.

**Phase 1.** Phase 1 is repeated.

**Guess.** The adversary outputs a guess \( \mu' \) of \( \mu \).

The advantage of an adversary in this game is defined as \( \Pr[\mu' = \mu] - 1/2 \).

**Definition 1.** eM-CP-ABE is secure in the selective-set model if all polynomial time adversaries have at most a negligible advantage in the selective-set game.

**Proof of security**

We prove that the security of eM-CP-ABE in the selective-set model reduces to the hardness of the DBDH assumption.

**Theorem 1.** If an adversary can break our scheme in the selective-set model, then a simulator can be constructed to play the DBDH game with a non-negligible advantage.

**Proof.** We suppose there exists a polynomial-time adversary \( A \) that can attack eM-CP-ABE in the selective-set model with advantage \( \epsilon \). Then, we can build a simulator \( \mathcal{B} \) that can play the DBDH game with advantage \( \epsilon/2 \). The simulation process is as follows:

First, the challenger provides two groups \( G_0, G_T \) with an efficient bilinear map, \( e \) and a generator \( g \). Then, the challenger throws a fair coin \( \mu \) out of view of \( B \). If \( \mu = 0 \), the challenger sets \( (A, B, C, Z) = (g^a, g^b, g^c, e(g, g)^{ab}) \); otherwise, the challenger sets \( (A, B, C, Z) = (g^a, g^b, g^c, e(g, g)^c) \). \( a, b, c, z \) are randomly chosen in \( \mathbb{Z}_p^* \).

**Init.** The simulator \( B \) runs a polynomial-time adversary \( A \). \( A \) chooses the \( 1 - n \) MAT \( T^* \) that it wishes to be challenged upon.

**Setup.** \( B \) chooses a random \( a' \in \mathbb{Z}_p \), and sets \( \alpha = a' - a + ab \). It calculates \( g_2 = e(g, g)^s = e(g, g)^{ab} \) and sets \( g_1 = g^b = B = g^b \). Then, it gives \( PK \) to \( A \).

**Phase 1.** \( A \) asks for the keys corresponding to any attributes set \( S(T^*(s)) = 0 \) from \( B \). \( B \) chooses a random \( r' \in \mathbb{Z}_p \) and sets \( r = r' + a - ab \). There exists \( D = g^{ab + r'} \). Then, we randomly choose \( r_j \in \mathbb{Z}_p^* \). \( \forall j \in S \). The left part of \( SK \) is \( D_j = g^{r' + a - ab} H_r(j)^s = Ag^{r - ab} H_r(j)^s \). \( D_j = g^{r'} \) gives the \( SK \) to \( A \).

**Challenge.** \( A \) submits two messages of equal length, \( M_0 \) and \( M_1 \), to \( B \). The challenger gets \( \mu \) by flipping a coin. Then, \( M_\mu \) is encrypted with \( T^* \). The ciphertext \( CT^* \) is sent to \( A \). In \( CT^* \), there exists \( B_0 = x_0 e(g, g)^{as}, C_0 = g^{b0} \). We set \( s_0 = s + c + (c/b), s \in \mathbb{Z}_p \), then we have

\[
C_0 = g^{b0} = g^{b(s + c + (c/b))} = g^{b(s + c)} g^{c} = g^{b(s + c)} C
\]

\[
B_0 = x_0 e(g, g)^{s + c + (c/b)} = x_0 T e(g, g)^\psi
\]

where \( \psi = (a' - a)(s + c + b/c) + abs + ac \).

**Phase 2.** The simulator repeats what it did in Phase 1.

**Guess.** \( A \) submits its guess \( \mu' \) of \( \mu \). If \( \mu' = \mu \), it indicates that \( A \) is given a valid BDH-tuple, \( (g^a, g^b, g^c, T = e(g, g)^{abc}) \); otherwise, it is given a random four-tuple \( (g^a, g^b, g^c, T = R) \).

If \( \mu' = \mu \), \( A \) finishes the guess game with the advantage of \( \epsilon \).

Thus

\[
Pr\left[ B\left(g^a, g^b, g^c, T = e(g, g)^{abc}\right) = 0 \right] = \frac{1}{2} + \epsilon
\]

Or else, the ciphertext \( CT^* \) is completely random. It means that \( A \) cannot get any information from \( CT^* \). Then

\[
Pr\left[ B(g, g^a, g^b, g^c, T = R) = 0 \right] = \frac{1}{2}
\]
The number of all related attributes is positively correlated to the cardinality of the attributes. Figure 10(a), the time taken to perform a bilinear operation is denoted as $T_{CP-ABE}$. The time taken to perform an exponential operation is denoted as $T_{CP-ABE}$. The time needed by CP-ABE, MCP-ABE, and eM-CP-ABE to satisfy the access control (Figure 10) is shown in Table 3. Compared with MCP-ABE, eM-CP-ABE has one more $P$, but $2l_0 - 1 - 2m_0$ less $E$. In the situation that $(2l_0 - 1 - 2m_0)E - P > 0$, eM-CP-ABE has better performance.

The theoretical analysis above proves that eM-CP-ABE consumes less encryption time than CP-ABE and MCP-ABE with the increase of $m_0$. The same conclusion is reached from the practical experiment (Figure 11). In this experiment, we assume that a group of access control policies exist that satisfy the following conditions:

1. $l = 3$;
2. $m_0 = \text{ceiling}(m/10)$.

Note that eM-CP-ABE is used as a digital envelope to encrypt the secret keys, not the content. Therefore, the size and resolution of videos have no effect on the efficiency of eM-CP-ABE. The three elements that affect eM-CP-ABE are $l$, $m$, and $m_0$. All experiments are implemented by C++ and run on a computer (Inter (R) Core (TM) i5-8300H CPU at 2.30 GHz, 8-GB RAM, Win-1064bit system).

In Figure 11, when $m = 10$, $m_0 = 1$, eM-CP-ABE and MCP-ABE consume similar time for encryption, slightly better than CP-ABE. With the increase of $m$, eM-CP-ABE has more advantages over MCP-ABE and CP-ABE in terms of the encryption time. For example, when $m = 100$, $m_0 = 10$, eM-CP-ABE consumes much less time than SCP-ABE for each attribute in the encryption and decryption processes.

We assume a $1-1$ MAT will encrypt one video with $l>1$ privilege levels and $m_0>0$ common attributes. The $i$th level at the hierarchical tree has $m_{i-1}$ leaf nodes (except the trunk). We can obtain $m_i = \sum_{i-1}^l m_{i-1}$. The number of all related attributes is $m = m_1 + m_0$. In Figure 10(a), the $1-1$ MAT has $l = 3$, $m_0 = 3$, $m_{1.1} = 3$, $m_{2.1} = 1$ and $m_{3.1} = 2$. To achieve the same access control, MCP-ABE needs a more complex tree (Figure 10(b)) and CP-ABE needs three trees (Figure 10(c)).

On the basis of Figure 10, we discuss the time and space complexity of KeyGen, Encryption, and Decryption of the three algorithms. The time taken to perform an exponential operation is denoted as $E$. The time taken to perform a bilinear operation is denoted as $P$.

**KeyGen**

For the three algorithms, there is the same attributes set $S$. The time complexity of executing KeyGen is positively correlated to the cardinality of the attributes set $|S|$, and the value is $(2|S| + 1)E$. The three algorithms have the same time complexity in KeyGen. Similarly, the private key lengths of the three algorithms are $(2|S| + 1)$. So, they have the same space complexity.

**Table 3. Time complexity of three encryption functions.**

| Algorithm       | Time complexity         |
|-----------------|------------------------|
| eM-CP-ABE       | $(l + 1)P + (l + 1 + 2m_0 + 2m_1)E$ |
| MCP-ABE         | $lP + (l + 2lm_0 + 2m_1)E$ |
| CP-ABE          | $T_{CP-ABE}$ |

**Encryption**

The time needed by CP-ABE, MCP-ABE, and eM-CP-ABE to satisfy the access control (Figure 10) is shown in Table 3. Compared with MCP-ABE, eM-CP-ABE has one more $P$, but $2l_0 - 1 - 2m_0$ less $E$. In the situation that $(2l_0 - 1 - 2m_0)E - P > 0$, eM-CP-ABE has better performance.

The theoretical analysis above proves that eM-CP-ABE consumes less encryption time than CP-ABE and MCP-ABE with the increase of $m_0$. The same conclusion is reached from the practical experiment (Figure 11). In this experiment, we assume that a group of access control policies exist that satisfy the following conditions:

1. $l = 3$;
2. $m_0 = \text{ceiling}(m/10)$.

Note that eM-CP-ABE is used as a digital envelope to encrypt the secret keys, not the content. Therefore, the size and resolution of videos have no effect on the efficiency of eM-CP-ABE. The three elements that affect eM-CP-ABE are $l$, $m$, and $m_0$. All experiments are implemented by C++ and run on a computer (Inter (R) Core (TM) i5-8300H CPU at 2.30 GHz, 8-GB RAM, Win-1064bit system).

In Figure 11, when $m = 10$, $m_0 = 1$, eM-CP-ABE and MCP-ABE consume similar time for encryption, slightly better than CP-ABE. With the increase of $m$, eM-CP-ABE has more advantages over MCP-ABE and CP-ABE in terms of the encryption time. For example, when $m = 100$, $m_0 = 10$, eM-CP-ABE

---

**Figure 10.** Different access trees under the same access rules: (a) eM-CP-ABE, (b) MCP-ABE, and (c) CP-ABE.
consumes 87 ms less time than MCP-ABE and 114 ms less than CP-ABE. The average encryption time per attribute is 0.79 ms, which is far less than 35 ms of SCP-ABE.

There are different ciphertext lengths in the three algorithms shown in Table 4. When $m_0 > 0$ and $l > 2$, the space complexity of the proposed algorithm is lower than M-CP-ABE and the value is $2(l + 1 + m_0) + 2(l - 1)m_0$.  

| Algorithm       | Space complexity |
|-----------------|------------------|
| eM-CP-ABE       | $2l_{max} + 2l_{max}$ |
| MCP-ABE         | $2l_{max} + 2(l - 1)m_0 + m_1$ |
| CP-ABE          | $SC_{CP-ABE}$    |

$SC_{CP-ABE} = 2l_{max} + 2l_{max} + 2m_0 + m_1 + \sum_{i=1}^{\infty} m_i$ is the space complexity of the proposed algorithm.

Decryption

We assume that the user has the second privilege in Figure 10, the MCP-ABE needs to iterate the DecryptNode function according to the access tree (Figure 12(a)), and the corresponding access structure for the eM-CP-ABE is Figure 12(b). They have the same decryption process. Let us consider another scenario. When the user wants to switch from a higher level privilege to lower level, CP-ABE needs to re-execute the decryption function. As the number of leaf nodes increases, so does the decryption time. When only one leaf node exists, the decryption time is about 15 ms, and when the number of leaf nodes is 10, the decryption time is 78 ms. The average computation time per 10 attributes of SCP-ABE is 25 ms. However, for eM-CP-ABE, the decryption time is a constant. To realize this switch of privilege, eM-CP-ABE only need to perform one and two hash operations, respectively. The hash operation takes much less time than the decryption function in ABE, about 3–8 ms, thus greatly reducing the decryption time.

Deployment

We deployed the streaming media system supporting HTTP live streaming (HLS) under the media framework proposed by Li et al.30 The system is shown in Figure 13. The video provider in User Area sends video and the corresponding access control policy to the media cloud. In media cloud, there are two areas: work and service. In work area, the whole video is divided into several segments; then, AES is used to encrypt the video segments with different keys; last, these keys are encrypted by eM-CP-ABE. In service area, authorization center, video storage and distribution are deployed. The video consumer in user area gets the video through the following steps:
1. Send the ABE key request to authorization center;
2. Get the SK;
3. Get the index file including the encrypted video segments keys from the video storage and distribution;
4. Decrypt the video segments keys using eM-CP-ABE;
5. Get the encrypted video segments from the video storage and distribution;
6. Decrypt the video segments using AES.

In this system, we implemented the decryption algorithm in Android version 4.2.2. After the two levels of decryption (ABE and AES), the plain-video segments are obtained. Then, the video segments are combined and played (Figure 14). In the full video playing process, the CPU consumption is stable (between 15% and 25%). It proves that the application based on our proposed algorithm can complete the video on-demand service smoothly.

Conclusion

Media cloud built by cloud providers collects the videos from video providers and provides kinds of video services with guaranteed quality to video consumers. Meanwhile, for video providers, they pay attention to whether the media cloud can provide effective copyright protection; for video consumers, they are concerned about whether the media cloud can respond to their request in real time. Therefore, how to establish a fine-grained and fast video access control scheme is a major challenge for media cloud.

In this article, eM-CP-ABE is proposed to meet the requirement of multi-level access control of massive videos in the mobile environment. Based on the MCP-ABE, the virtual node and key chain are introduced to construct a new hierarchical access control structure. In the new structure, common attributes are extracted to the root node to save the encryption cost. When the user switches from a higher level privilege to a lower level, only a few hash operations are required without re-calling decryption in ABE, which greatly saves the decryption time and computation consumption.

In addition, we discussed the extended application of the proposed algorithm. Multi-authorization can improve the scalability of the algorithm. Using integer comparison, user revocation can be realized. More important, the access tree in eM-CP-ABE can be extended to $1 - n$ MAT that improves the efficiency of the access control structure and saves the cost of tree generation and maintenance. We proved that eM-CP-ABE can successfully resist CPAs under the DBDH assumption. Performance analysis and experiment show that the proposed algorithm is better than MCP-ABE and CP-ABE in the encryption process.

Acknowledgements

The authors thank the anonymous reviewers for their very helpful comments which helped to improve the presentation of this article.

Declaration of conflicting interests

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

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: The research was funded by China Postdoctoral Science Foundation 2019M663358, Fundamental Research Funds for the Central Universities of the Communication University of China (CUC2019A009), the High-quality and Cutting-edge Disciplines Construction Project for Universities in Beijing (Internet Information, Communication University of China; CUC190J054), and National Key Task Team Cultivation Project 2019 (CUC19ZD003).

ORCID iD

Hao Li https://orcid.org/0000-0002-9045-561X

References

1. Chen H, Bhargava B and Zhongchuan F. Multilabels-based scalable access control for big data applications. *IEEE T Cloud Comp* 2014; 1(3): 65–71.
2. Yeh L-Y, Chiang P-Y, Tsai Y-L, et al. Cloud-based fine-grained health information access control framework for lightweight IoT devices with dynamic auditing and attribute revocation. *IEEE T Cloud Comp* 2018; 6(2): 532–544.
3. Li W, Liu BM, Liu D, et al. Unified fine-grained access control for personal health records in cloud computing. *IEEE J Biomed Health* 2019; 23: 1278–1289.
4. Chaudhary R, Aujla GS, Garg S, et al. SDN-enabled multi-attribute-based secure communication for smart grid in IoT environment. *IEEE T Ind Inform* 2018; 14(6): 2629–2640.

5. Guan Z, Li J, Zhu L, et al. Toward delay-tolerant flexible data access control for smart grid with renewable energy resources. *IEEE T Ind Inform* 2017; 13(6): 3216–3225.

6. Hong J, Xue K, Xue Y, et al. TAFC: time and attribute factors combined access control for time-sensitive data in public cloud. *IEEE T Serv Comput* 2020; 13: 158–171.

7. Wu Y, Wei Z and Deng RH. Attribute-based access to scalable media in cloud-assisted content sharing networks. *IEEE T Multimedia* 2013; 15(4): 778–788.

8. Ma C, Yan Z and Chen W. Scalable access control for privacy-aware media sharing. *IEEE T Multimedia* 2019; 21: 173–183.

9. Bethencourt J, Sahai A and Waters B. Ciphertext-policy attribute-based encryption. In: *Proceedings of the 2007 IEEE symposium on security and privacy (SP’07)*, Berkeley, CA, 20–23 May 2007, pp. 321–334. New York: IEEE.

10. Li J, Zhang Y, Chen X, et al. Secure attribute-based data sharing for resource-limited users in cloud computing. *Comput Secur* 2018; 72: 1–12.

11. Cui J, Zhou H, Xu Y, et al. OOABKS: online/offline attribute-based encryption for keyword search in mobile cloud. *Inform Sciences* 2019; 489: 63–77.

12. Han J, Susilo W, Mu Y, et al. Improving privacy and security in decentralized ciphertext-policy attribute-based encryption. *IEEE T Inf Foren Sec* 2015; 10(3): 665–678.

13. Xue K, Xue Y, Hong J, et al. RAAC: robust and auditable access control with multiple attribute authorities for public cloud storage. *IEEE T Inf Foren Sec* 2017; 12(4): 953–967.

14. Xue Y, Xue K, Gai N, et al. An attribute-based controlled collaborative access control scheme for public cloud storage. *IEEE T Inf Foren Sec* 2019; 14: 2927–2942.

15. Yuen TH, Liu JK, Man HA, et al. k-times attribute-based anonymous access control for cloud computing. *IEEE T Comput* 2015; 64(9): 2595–2608.

16. Xu S, Yang G and Mu Y. Revocable attribute-based encryption with decryption key exposure resistance and ciphertext delegation. *Inform Sciences* 2019; 479: 116–134.

17. Wei J, Liu W and Hu X. Secure and efficient attribute-based access control for multiauthority cloud storage. *IEEE Systems Journal* 2018; 12(2): 1731–1742.

18. Xue L, Yu Y, Li Y, et al. Efficient attribute-based encryption with attribute revocation for assured data deletion. *Inform Sciences* 2019; 479: 640–650.

19. Li J, Yao W, Zhang Y, et al. Flexible and fine-grained attribute-based data storage in cloud computing. *IEEE T Serv Comput* 2017; 10(5): 785–796.

20. Ning J, Dong X, Cao Z, et al. White-box traceable ciphertext-policy attribute-based encryption supporting flexible attributes. *IEEE T Inf Foren Sec* 2015; 10(6): 1274–1288.

21. Ning J, Cao Z, Dong X, et al. White-box traceable CP-ABE for cloud storage service: how to catch people leaking their access credentials effectively. *IEEE T Depend Secure* 2018; 15(5): 883–897.

22. Wang S, Zhou J, Liu JK, et al. An efficient file hierarchy attribute-based encryption scheme in cloud computing. *IEEE T Inf Foren Sec* 2016; 11(6): 1265–1277.

23. Li J, Chen N and Zhang Y. Extended file hierarchy access control scheme with attribute based encryption in cloud computing. *IEEE T Emerg Top Comput*. Epub ahead of print 12 March 2019. DOI: 10.1109/TETC.2019.2904637.

24. Bobba R, Khurana H and Prabhakaran M. Attribute-sets: a practically motivated enhancement to attribute-based encryption. In: *Proceedings of the European symposium on research in computer security*, Luxembourg, 23–27 September 2009, pp. 587–604. Cham: Springer.

25. Wang Z, Liu J, Zhang R, et al. A collusion-resistant conditional access system for flexible-pay-per-channel pay-TV broadcasting. *IEEE T Multimedia* 2013; 15(6): 1353–1364.

26. Wang Z, Liu J and Deng RH. HASBE: a hierarchical attribute-based solution for flexible and scalable access control in cloud computing. *IEEE T Inf Foren Sec* 2012; 7(2): 743–754.

27. Teng W, Yang G, Xiang Y, et al. Attribute-based access control with constant-size ciphertext in cloud computing. *IEEE T Cloud Comp* 2017; 5(4): 617–627.

28. Yang K, Liu Z, Jia X, et al. Time-domain attribute-based access control for cloud-based video content sharing: a cryptographic approach. *IEEE T Multimedia* 2016; 18: 940–950.

29. NIST special publication 1800-3 attribute based access control, [https://csrc.nist.gov/publications/detail/sp/1800-3/archive/2015-09-29](https://csrc.nist.gov/publications/detail/sp/1800-3/archive/2015-09-29)

30. Li H, Yang C and Liu J. A novel security media cloud framework. *Comput Electr Eng* 2019; 74: 605–615.