IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received May 22, 2018, accepted July 3, 2018, date of publication July 9, 2018, date of current version July 30, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2854600

Efficient Compressed Ciphertext Length Scheme Using Multi-Authority CP-ABE for Hierarchical Attributes

ZHIYONG ZHANG[1], (Senior Member, IEEE), CHENG LI[1], BRIJ B. GUPTA[2], (Senior Member, IEEE), AND DANMEI NIU[1]

[1]Information Engineering College, Henan University of Science and Technology, Luoyang 471023, China
[2]National Institute of Technology Kurukshetra, Kurukshetra 136119, India

Corresponding author: Zhiyong Zhang (xidianzzy@126.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 61772174 and Grant 61370220, in part by the Plan For Scientific Innovation Talent of Henan Province under Grant 174200510011, in part by the Program for Innovative Research Team (in Science and Technology) in the University of Henan Province under Grant 15IRTSTHN010, in part by the Program for Henan Province Science and Technology under Grant 142102210425, in part by the Natural Science Foundation of Henan Province under Grant 162300410094, and in part by the Project of the Cultivation Fund of Science and Technology Achievements of Henan University of Science and Technology under Grant 2015BZCG01.

ABSTRACT In an attribute-based encryption, the user is identified with help of some attributes and their functions for encryption and decryption of the data. The current techniques based on attribute-based encryption have found that if user's access structure includes a considerable amount of attribute information labeled as *Don't Care*, then the encryption pairing operation has low calculation efficiency and ciphertext information redundancy. In this paper, we have proposed a hierarchical multi-authority attribute-based encryption on prime order groups to tackle these problems. Our encryption technique has a polycentric attribute authorization system based on an AND gate access structure, with a unified attribute index established by each attribute authority throughout the system, to form a binary tree, i.e., attribute access tree. The state value of the parent node can be determined by the state of its child node in an attribute access tree. The attribute-based encryption established in this manner is theoretically proven to effectively decrease the calculation amount for decryption and compress the redundant information in the ciphertext as much as possible. Our encryption technique has a theoretical and practical significance in the system of ''large universe'' constructions.

INDEX TERMS Hierarchical attribute-based encryption, large-universe, multi-authority ciphertext-policy attribute-based encryption, redundant information removal.

I. INTRODUCTION

With the evolution of web and social media websites more and more personal data stored over such sites need to be secured. The classical cryptosystems use keys which need to be shared with the other party for encryption or decryption. As one of modern cryptography algorithms, attribute-based encryption (ABE) has been widely applied and has gained extensive attention in fields such as cloud storage, social network, and on-demand online service. Compared to the classical encryption algorithms, ABE enabled fine-grained access control and made the development of a more flexible access control structure possible.

In 2005, Sahai and Waters [1] proposed a fuzzy identity-based encryption (FIBE) algorithm. This technique considers a packet of a particular user's identity information as the input of data encryption and decryption. Only users conforming to the access control structure developed by the data owner have access to the data information. Thus, this technique makes the achievement of point-to-multipoint communication feasible, as it was impossible for previous encryption algorithms. In addition, the data owner can develop a flexible authorization structure according to a series of identity information of a data receiver (user attribute).

In 2006, key-policy attribute-based encryption (KP-ABE) and ciphertext-policy attribute-based encryption (CP-ABE) were proposed for the first time by Goyal *et. al.* [2]. In 2007, Bethencourt *et al.* [3] proposed the scheme which was conceptually similar to Role-based access control where

2169-3536 © 2018 IEEE. Translations and content mining are permitted for academic research only.
Personal use is also permitted, but republication/redistribution requires IEEE permission.
See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

data confidentiality is ensured even with untrusted servers. At the same time, the first CP-ABE with an AND gate access structure by Cheung and Newport [4]. After that, Sun *et al.* [5] proposed a method based on AND gate verifiable attribute-based keyword search, and does not depend on a always online trusted authority, and is safe and efficient.

However, the ABE techniques discussed above are all single-authority ABEs. With the rapid increase of web communication capacity, single-authority gradually fails to meet the needs of users due to its low efficiency. Therefore, Chase [6] proposed a multi-authority attribute-based encryption (MA-ABE), in which multiple Attribute Authorities (AAs) emerge to administer user's attributes. In MA-ABE, the system allots a global identifier (GID) to every user, which is connected to the attribute key issued by AAs, in order to prevent a collusion attack. The possible emergence of one Central Authority (CA) [6]–[9], or a number of CAs, or no CA [10]–[12] in the MA-ABE is worthy of noting. In particular, as CA has the Master Key (MK) of the whole system, the MK will be divulged upon an attack, which endangers the security of the whole system. But most of the researchers looked at the constant ciphertext length of lightweight encryption algorithm. However, it is also an urgent problem to effectively reduce the cipher redundant information in the "big universe" attribute cryptosystem, in order to reduce the network pressure and improve the storage efficiency. More related work will be pointed out in the subsequent chapters.

Based on the above work, this paper proposes a MA CP-ABE scheme using AND-gate to solve the problem of authorization in the system of "big universe," which makes it more suitable for practical application. This solution can effectively reduce the computational expense and decrease the ciphertext's length when users build sparse attribute access structures (For example, the length of ciphertext will increase and the efficiency of decryption will be reduced when the user specifies a lot of "Don't Care" information. Therefore, the "Don't Care" attribute information needs to be compressed as much as possible to optimize the system efficiency.). A number of CAs and AAs emerge in the system. Each AA indexes the attributes it administers uniformly (the specific structure will be introduced below, thus not explained here) for the access structure to have increased flexibility and for the attributes in each AA to be added according to requirements. In addition, computation efficiency is effectively increased in generating and decrypting ciphertexts. Moreover, this solution adopts a polycentric (non-center) model, which provides a high security lever for the system. The contributions of this paper are as follows: 1) Using hierarchical attributes, the redundant information in the ciphertext is compressed to improve the system efficiency and save storage space; 2) The solution is implemented in a no-central environment, effectively increasing the security; 3) This solution is well compatible with the "large universe" attribute cryptosystem.

The remaining of the paper is organized as follows. The first section introduces the research results of the scholars in this paper, including the development process of ABE and the application of ABE. The second section briefly introduces some definition of system. The third section proposes the scheme formalization and security model. Next, we present the corresponding specific algorithm. The safety and performance analysis is carried out in detail in the fifth section. At the final section, we summarize the works of this paper.

II. RELATED WORK

With the development of the network and the continuous increase in the amount of data, the lack of performance of a single authority has become more apparent. More and more experts and scholars have begun to pay attention to the attribute encryption scheme under multiple authorizations.

In [13] a Multi-CA CP-ABE is (was) presented, which effectively increased the calculation efficiency of Single-CA and tackled the security problem brought by the Single-CA controlling the MK. However, this solution is founded on Composite order groups, which has a huge computation overhead. Reference [14] puts forward a solution that builds a multi-attribute authority on prime order groups, which effectively reduced the computation overhead. In [15], an attribute-based friend discovery system was constructed using Shamir's threshold secret sharing and CP-ABE. In the system, Trusted Authority (TA) manages a hierarchical AAs, but must ensure that the TA is absolutely trustworthy and that it is limited by the size of the attribute set. In [16] and [17], the length of a ciphertext is fixed and controllable, and the performance of the system is improved, the proposed solution only applies to limited attribute encryption systems. In [18] a decentralized multi-authority ABE is given, with multiple CAs and AAs. In this solution, the identity key of users is cooperatively authorized by multiple CAs. However, the authorization access structure in the ciphertext is hidden; therefore, the attributes of all attribute authorities must be calculated, which ensures the constant length of the ciphertext, which is hard to achieve in the system of "large universe" constructions. In [19], the use of the file is hierarchical encryption, effectively enhance the encryption efficiency, the idea of a certain reference, but the program on the access control policy update is a challenge. In [20], an authority that permits the existence of multiple independent authorities and any polynomial number is put forward to monitor attributes. The issuing of keys and decryption information can be decrypted rapidly without the online solutions of all authorities. In [21], according to the previous MA-ABE program proposed a more flexible encryption authorization system, in the system without all the certification body online, and the user's key and GID binding to defend against conspiracy attacks. At the same time the authority of the system can work independently, making the system more flexible.

Although most researchers have done a lot of research on the length of the constant ciphertext, there are two problems: 1) they can't meet the needs of the "large universe" system;

TABLE 1. Related work analysis.

	Access Structure	Length of Ciphertext	Application Scenario	Multi-Authority Scheme
[4]	AND Gate	Non-constant	Lightweight/ Large Universe	NO
[8]	LSSS	Non-constant	Large Universe	Single-CA
[9]	(t,n) Threshold	Non-constant	Large Universe	Single-CA
[15]	Tree	Non-constant	--	Single-CA
[17]	AND Gate	Constant	Lightweight	NO
[18]	AND Gate	Constant	Lightweight	Multi-center/ CAs
[21]	AND Gate	Non-constant	--	Multi-center/ CAs
Our	AND Gate	Non-constant	Large Universe	Multi-center/ CAs

2) the ciphertext contains some redundant information; and so on. The next part of the relevant research will be compared in the form of a table, as shown in Table 1:

III. PREPARATION

A. BLIINEAR LOGARITHM

Assume that G and G_T are two multiplication cyclic groups, having its order as a prime number p, g is a generator of Group G, and thus, a bilinear map: $e : G \times G \rightarrow G_T$ exists with the following properties:

1) Bilinearity: for any $u, v \in G; a, b \in \mathbb{Z}_p$, $e(u^a, v^b) = e(u, v)^{ab}$

2) Non-degeneracy: for calculation, $e(g, g) \neq 1$

3) Symmetry: e(,) is a symmetry operation, i.e., $e(g^a, g^b) = e(g^b, g^a) = e(g, g)^{ab}$

B. ACCES POLICY

Definition 1 (Access Structure [22], [23]): Assuming that $\mathbb{P} = \{P_1, P_2, \ldots, P_n\}$ is a participant set, $\mathbb{AS} \in 2^{\mathbb{P}}$, $\mathbb{AS}$ an authorized set, and $2^{\mathbb{P}}$ the attributes set of all $\mathbb{P}$ subsets. If $\forall A, B$ conforms to $A \in \mathbb{AS}$, $A \subseteq B$ and $B \in \mathbb{AS}$, then $\mathbb{AS}$ is a monotonic access structure. An access structure conforming to $\mathbb{AS}$ is called an authorized set, whereas those that do not conform to $\mathbb{AS}$ are non-authorized sets.

Definition 2 (AND-Gate Access Structure [4]): Assume that $\mathbb{M} = \{1, 2, \ldots, m\}$ is the attribute in the global attribute domain in the number of m, and for attribute $i \in \mathbb{M}$, text i exists representing its semantics in the access structure. Text i has three representations: $+i$ (positive), $\neg i$ (negative), and $\bar{i}$ (*Don't Care*). Assume that the access structure is $W = \wedge_{i\in\mathbb{M}} \underline{i}$ if user set $\mathbb{S}$ conforms to access structure W, $\mathbb{S} \models W$; otherwise, $\mathbb{S} \not\models W$.

Definition 3: In the structure of hierarchical attribute access tree Γ: each attribute authority encodes the attributes in the domain, offering every attribute its own index, and then establishes a binary tree. In the non-leaf node $T_{[i\ldots j]}$ of this attribute tree, the initial sequence number is the index value of the leftmost leaf node below this non-leaf node; the closing sequence is the index value of the rightmost leaf node of this non-leaf node. In leaf node T_{2n+i}, n represents the number of attributes included in the attribute access tree. In particular, when the number of attributes is uneven, a null attribute of which the expression value is $\bar{i}$ will constantly be supplemented on the rightmost side of the binary tree to maintain the balance of all non-leaf child nodes in the attribute tree. The attribute access tree established for a particular attribute authority k is defined as Γ_k.

TABLE 2. Important abbreviations and their meanings.

Symbol	Meaning
MK	Master key
CT	CEK Encrypted ciphertext
$\mathbb{I}, \mathbb{M}, \mathbb{S}$	Access authorization attributes set, global attribute domain attribute collection, user personal attribute collection
SK	User private key, including attribute private key SK_{AA} and Central Authority Identity Private Key SK_{CA}
M	Attribute encryption plaintext
$\mathcal{A}$	A adversary

C. SYMBOL DESCRIPTION

D. INTRACTABILITY ASSUMPTION

It determines the bilinear Diffie–Hellman (DBDH) problem: in the bilinear group G, established in 1.1, of which the order is prime number p, $\mathbb{Z}_p$ is a finite domain, $a, b, c, h \in \mathbb{Z}_p$. Establish two four-tuple: $(g^a, g^b, g^c, (g, g)^{abc})$ and $(g^a, g^b, g^c, (g, g)^h)$.

The adversary $\mathcal{A}$ cannot distinguish the above mentioned two tetrads with a non-negligible advantage in one polynomial time. The advantage of adversary $\mathcal{A}$ is defined as follows:

$$Adv_{\mathcal{B}}^{DBDH} = \left| \begin{array}{l} \Pr[\mathcal{B}(g^a, g^b, g^c, (g, g)^{abc}) = 1] \\ - \Pr[\mathcal{B}(g^a, g^b, g^c, (g, g)^h) = 1] \end{array} \right|$$

IV. SCHEME FORMALIZATION AND SECURITY MODEL

A. CONCEPTS OF THE SCHEME FORMALIZATION

As shown in Fig.1, this scheme adopted by the overall architecture can be roughly divided into four parts: 1) the data owner, 2) data receiver, 3) data storage side, 4) authorized institution (including a number of central authority and a number of properties authorized institution). The data owner submits the access control policy to the authorization mechanism; The authorizer generates the required material for encryption and sends it to the data owner; The data owner encrypts the text and stores the ciphertext information on the data storage side according to the material sent by the authorized institution. Receiver when there is data from the data store side ciphertext information, and the user's keys will be issued by authorized institution for the information, when the data of the receiver does not meet the specified data owners access control policy, will not be able to decrypt the ciphertext information correctly. The above is the encryption and decryption authorization process for the data of the scheme, and the scheme will be formally introduced below.

In a hierarchical multi-attribute authorization model, several $CA = \{CA_1, CA_2, \ldots, CA_l\}$ exist, l is the number of CAs; several $AA = \{AA_1, AA_2, \ldots, AA_t\}$, t is the number of AAs; and several u (user)s emerge. In a system, each user is allocated an exclusive identifier GID. Central authority allocates identity keys relevant to the GID of users; AA offers user attribute-related keys. Each AA_j administers an attribute domain U_j and with any AA_i, $U_i \cap U_j = \varnothing$, which makes every attribute to be administered by one particular AA exclusive and the attributes administered by each AA coexist with one another. The global attributes set $U = \cup_{j=1}^{t} U_j = \{1, 2, \ldots, m\}$ is special because the attributes in the global domain, in the number of m, have an exclusive index value, and the index in each attribute is also exclusive. For better comprehension, all nodes of an attribute access tree are expressed as $T_{k,[\ldots]}$, attribute authority AA_k as k, and information of the nodes on the access tree established by this attribute authority as $[\ldots]$.

The solution in this paper contains seven algorithms, of which the constitution is described as follows:

1. $setup(1^\lambda \rightarrow \delta)$: Takes input as the safe random parameter 1^λ in the system, and the system outputs the global public parameter δ, a safe signature algorithm $\Sigma_{sign} = (KenGen, Sign, Verify)$ and the system's master key parameter MK.
2. $CAsetup(\delta, d \rightarrow CPK_d, CSK_d, CAPV_d)$: Takes input as the global public parameter δ and the index of every CA adopting the initialization algorithm, and the system outputs identity public parameter CPK_d, the private identification key CSK_d of authority CA_d, and CA_d's corresponding verification signature $CAPV_d$.

$AAsetup(\delta, k \rightarrow T_{k,i}, t_{k,i})$: Operates the initial algorithm with every AA by inputting the global public parameter δ; for AA_k, output attribute public parameter $\cup T_{k,i}$, attribute implicit parameter $\cup t_{k,i}$, and AA_k's corresponding public random parameter r_k.

3. $Encryption(\delta, M, W \rightarrow CT)$: Input message M, the global public parameter δ, and access structure W designated by the user; adopt the encryption algorithm for attribute access tree Γ established by traversing the attribute authority; and obtain the user's enciphered data CT.
4. $CAKeyGen(D', GID \rightarrow D, D^*, \Theta, \Xi)$: The user offers CA_d a GID, CA_d will input the parameter r generated by AA and the user's exclusive identifier GID and then outputs the private identification key $SK_{CA_d} = (D, D^*, \Theta)$and the attribute key generation parameter Ξ.
5. $AAKeyGen(r, \cup t_{k,i}, GID, \Xi \rightarrow D_i, F_i)$: The user u_{GID} applies to AA_k for a private attribute key; AA_k will then verify the identity information offered by the user. If the information is genuine, then AA_k will input the allocated r_k and $\cup t_{k,i}$ and will output a set of private attribute key $SK_{AA} = (\{\langle D_i, F_i\rangle \mid i \in \mathbb{M}\})$ for the user. As the user only has to own the private attribute key corresponding to his/her own attributes set, the key can only contain the index information of the attributes. If not genuine, then the issuing of attribute key will be terminated.
6. $Decryption(D^*, SK, CT \rightarrow M)$: The authorized user inputs the private attribute key SK_{AA}, private identification key SK_{CA}, and ciphertext CT provided by the system, and then the system operates the encryption algorithm according to the information offered by the user. If the user conforms to the access structure, then M becomes the input; otherwise, $\perp$ and the process will be terminated.

B. SECURITY MODEL

In the security confirmation process, a Challenger and an Adversary are defined. The Adversary chooses and challenges a Challenger, and the chosen Challenger accepts this challenge to play an indistinguishable under chosen plaintext attack (IND-CPA) game.

The rules of an IND-CPA game are as follows:

1) System initialization: A challenger inputs a random parameter λ, and then the system's public parameter δ and master key parameter MK are generated.
2) Phase I: The Adversary presents the attributes set $\mathbb{S} = (att_1, \ldots, att_m)$ to the Challenger, which it wants to inquire and its exclusive identifier GID, and then the Challenger generates the corresponding private attribute key and identification key by operating the key generation algorithm and presents them to the Adversary.
3) Challenging phase: The Adversary chooses and presents to the Challenger two messages M_0 and M_1 with the same length and an authorized access set $\mathbb{Q}$, which it wants to challenge. $\mathbb{S} \cap \mathbb{Q} = \varnothing$ is worthy of note. The Challenger randomly chooses $b \in \{0, 1\}$, computes the ciphertext $CT_b = Encryption(M_b, \delta, \mathbb{Q})$, and presents the latter to the Adversary.
4) Phase II: The Adversary repeats the work in Phase I and continues to inquire the private attribute key of an attributes set $\mathbb{O} = (att_{m+1}, \ldots, att_n)$. $\mathbb{O} \cap \mathbb{Q} = \varnothing$ is worthy of note. The Challenger computes the private

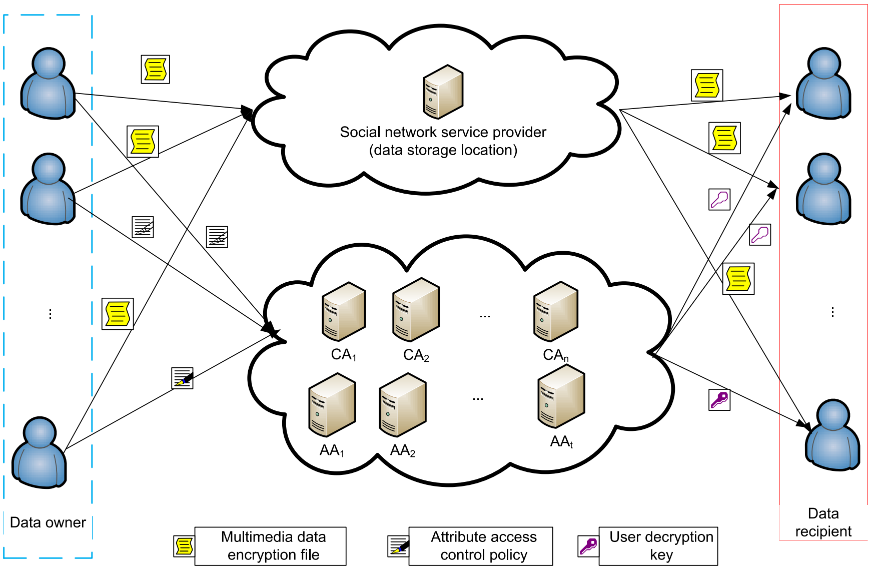


FIGURE 1. System overall structure diagram.

attribute key according to the attribute value in the attributes set $\mathbb{O}$and presents this attribute key to the Adversary.

5) Guess phase: The Adversary inputs his conjecture about $b' \in \{0, 1\}$ according to the information in hand. If $b = b'$, then the Adversary wins.

The advantage probability of the Adversary's winning is defined as $\varepsilon := \left|\Pr[b' = b] - 1/2\right|$

V. DESIGN OF THE SCHEME CONSTRUCTION

According to [13], if the solution is adaptively secure, then the number of CAs in a trusted environment will not influence system security. Despite, a single CA has the ability to encrypt all the ciphertexts in the system. Therefore, to better comprehend the system algorithm dealing with single-CA systems, it is described in this paper in addition to security analysis. The key generation algorithm for multi-CA systems will be dealt with later.

A. CONSTRUCTION FOR SINGLE-CA SYSTEMS

Setup: Input a security parameter 1^λ in the system, and then the system will generate two multiplication cyclic groups G and G_T, of which the order is prime number p, which adopts the group generation algorithm. Group G's generator is g. Meanwhile, there is a chosen signature algorithm$\Sigma_{sign} = (KenGen, Sign, Verify)$, whose existence cannot be fabricated, and then output the global public parameter $\delta = (e, g, G, G_T, p, \Sigma_{sign})$.

CASetup: Choose a random number $y, \mu \in \mathbb{Z}_p$, assume $Y = e(g, g)^y$, $CPK = Y$ as the public encrypting parameter and $CSK = y$ as CA's master key, and uses the signature algorithm to obtain $VerifyKey = \mu$, $CAPV = (SignKey, \mu)$.

AASetup: The global attribute authority randomly selects $r, z_1, z_2, \ldots, z_{3m} \in \mathbb{Z}_p$, in which m represents the number of attributes included in the global attribute domain; meanwhile, for the non-leaf nodes $x = [i \ldots j]$ in the attribute authority AA_k, randomly selected $z_{k,x} \in \mathbb{Z}_p$ and $Z_{k,x} = g^{z_{k,x}}$ exists. At the same time, r $= \sum_{k=1}^{t} r_k$, where $k = (1, 2, \ldots, t)$ is the index of attribute authorities.

For better comprehension, the attributes in the system in this paper are indexed. For a single AA, the index can be used independently and without repetition or the need to know the existence of the other attribute authorities; attribute authorization can be operated independently. $Z_i = g^{z_i}$ exists, in which $Z_{\mathrm{i}}, Z_{n+i}, Z_{2n+i}$ exists, corresponding to $+i$(Positive), $-i$(Negative), $Don'tCare$, respectively. Therefore, the public attribute parameter $\cup Z_{k,i} = Z_{k,1}, \ldots, Z_{k,3n}$ and the implicit attribute parameter $\cup z_{k,i} = z_{k,1}, \ldots, z_{k,3\mathrm{n}}$, where $k = (1, 2, \ldots, t)$ is the index of attribute authorities.

Encryption: Set plaintext M and select first-time encrypted random parameter s $\in \mathbb{Z}_p$, and then the encrypted plaintext $C = M \cdot Y^s$ and $C' = g^s$ are generated. The user establishes an access control structure $\mathbb{I} \in \mathbb{M}$ ($\mathbb{M}$ is the global attributes set) and sets an access control structure $W = \wedge_{i \in \mathbb{I}} \underline{i}$.

In the present study, a recursive algorithm Traversal is defined; a balanced binary tree has been defined previously; thus, every internal node of the binary tree includes two child nodes.

1) First, define $att_{k,i}$ and $att_{k,i+1}$ as two leaf nodes, of which the corresponding attribute values are i and $i + 1$.

- If for every $i \in \mathbb{I}$ or $i + 1 \in \mathbb{I}$, the following equation exists: if $\underline{i} = +i$, $C''_{k,i} = Z^s_{k,i}$; if $\underline{i} = -i$, $C''_{k,i} = Z^s_{k,n+i}$; and if $(i \in \mathbb{M} \backslash \mathbb{I}) \wedge (i + 1 \in \mathbb{I})$, $C''_{k,i} = Z^s_{k,2n+i}$. Meanwhile, if the relationship is contained in the ciphertext, then return to 1;
- If $i \notin \mathbb{I} \wedge i + 1 \notin \mathbb{I}$, then return to 0.

2) If $att_{k,i}$ and $att_{k,i+1}$ are two non-leaf nodes, then operate Traversal, of which the rules are as follows:

- If two non-leaf nodes $att_{k,i}$ and $att_{k,i+1}$ both return to 1, then return to 1;
- If two non-leaf nodes $att_{k,i}$ and $att_{k,i+1}$ both return to 0, then return to 0;
- If $att_{k,i}$ returns to 0 and $att_{k,i+1}$ returns to 1, then define $C''_{k,att_i} = T^s_{k,att_i}$ and write C''_{k,att_i} and its corresponding node information into the ciphertext, and then return to 1.

Lastly, the encrypted ciphertext $CT = (W, C, C', \{C'' = \Pi C''_{k,i\in\mathbb{M}}\})$ is written.

CAkeygen: According to the user's exclusive identifier, randomly select for $o_{GID} \in \mathbb{Z}_p$ for him; hence, $D = e(g, g)^{o_{GID}}$, $D^* = g^{y-r}$, $\Theta = 1/o_{GID}$, and then get CA's private identification key $SK_{CA} = (D, D^*, \Theta)$. It operates the signature algorithm, sign $(Signkey, GID)$, and then gets ϑ_{GID} and the generation parameter of the user's attribute key $\Xi = (\vartheta_{GID}, GID)$, which is only used for the AA generating attribute key.

AAKeyGen: On receiving an attribute authorization request, AA_k it extracts the verification information $CAPV$ in Ξ at the beginning. If verification fails, then $\perp$ will be the output. Assume that $r_{GID} \in \mathbb{Z}_p$ is a random parameter and $r = r_{GID} = \sum_{i,k=1}^{i=m,k=t} r_{GID,k,i}$, $r_{GID,k,x} = \Sigma r_{GID,k,i}$ (i represents all of x's child nodes). Assume that the user's attributes set is $\mathbb{S}$ and that $i \in \mathbb{M}$ exists, if and only if $i \in \mathbb{S}$, $\underline{i} = +i$; when $i \notin \mathbb{S}$, $\underline{i} = -i$. When the attribute $i \in \mathbb{M}$ and $i \in \mathbb{S}$, assume that $D'_{GID,k,i} = g^{(r_{GID,k,i}/z_{k,i})\cdot o_{GID}}$; when the attribute $i \in \mathbb{M}\backslash\mathbb{S}$, assume $D'_{GID,k,i} = g^{(r_{GID,k,i}/z_{k,n+i})\cdot o_{GID}}$; when $i \notin \mathbb{M}$, assume $F'_{GID,k,i} = g^{(r_{GID,k,i}/z_{k,2n+i})\cdot o_{GID}}$; for non-leaf nodes, $F'_{GID,k,x} = g^{(r_{GID,k,x}/z_{k,x})\cdot o_{GID}}$ exists. Lastly, the private attribute key $SK_{AA} = (\{\left\langle D'_{k,i}, F'_{k,i}\right\rangle | k = \{1 \ldots t\}, i \in \mathbb{M}\})$ is obtained.

Decryption: Input ciphertext $CT = (W, C, C', \{C''_{i\in\mathbb{M}}\})$, private attribute key $SK_{AA} = (\left\{\left\langle D'_{k,i}, F'_{k,i}\right\rangle | k = \{1 \ldots t\}, i \in \mathbb{M}\right\})$, and CA's private identification key $SK_{CA} = (D, D^*)$. Then, it extract access control structure $W = \wedge_{i\in\mathbb{I}} \underline{i}$ from CT, determine user attributes set $\mathbb{S}$, and access control structure W; if $\mathbb{S}$ does not conform to W, then $\perp$ will be the output and the encryption will be terminated; otherwise, all attributes $i \in \mathbb{I}$ will be encrypted in the following process:

1) If node x is a leaf node:

- If attribute $i \in \mathbb{S}$ and attribute setting $\underline{i} = +i$, then

$$\begin{aligned} N_1 &= \Pi(e(C''_{k,i}, D'_{k,i}) \\ &= \mathrm{e}(Z^s_{k,i}, g^{(r_{GID,k,i}/z_{k,i})\cdot u_{GID}}) \\ &= \mathrm{e}(g^{z_{k,i}\cdot s}, g^{(r_{GID,k,i}/z_{k,i})\cdot u_{GID}}) \\ &= \mathrm{e}(\mathrm{g}, \mathrm{g})^{s\cdot r_{GID,k,i}\cdot u_{GID}}) \end{aligned}$$

- If attribute $i \notin \mathbb{S}$ and attribute setting $\underline{i} = -i$, then

$$\begin{aligned} N_2 &= \Pi(e(C''_{k,i}, D'_{k,i}) \\ &= \mathrm{e}(Z^s_{k,n+i}, g^{(r_{GID,k,i}/z_{k,n+i})\cdot u_{GID}}) \\ &= \mathrm{e}(g^{s\cdot z_{k,n+i}}, g^{(r_{GID,k,i}/z_{k,n+i})\cdot u_{GID}}) \\ &= \mathrm{e}(\mathrm{g}, \mathrm{g})^{s\cdot r_{GID,k,i}\cdot u_{GID}}) \end{aligned}$$

- If attribute setting $\underline{i}$ is $Don'tCare$, then

$$\begin{aligned} N_3 &= \Pi(e(C''_{k,i}, D'_{k,i}) \\ &= \mathrm{e}(Z^s_{k,2n+i}, g^{(r_{GID,k,i}/z_{k,2n+i})\cdot u_{GID}}) \\ &= \mathrm{e}(g^{s\cdot z_{k,2n+i}}, g^{(r_{GID,k,i}/z_{k,2n+i})\cdot u_{GID}}) \\ &= \mathrm{e}(\mathrm{g}, \mathrm{g})^{s\cdot r_{GID,k,i}\cdot u_{GID}}) \end{aligned}$$

2) If node x is a non-leaf node:

$$\begin{aligned} N_4 &= \Pi(e(C''_{k,i}, D'_{k,i}) \\ &= \mathrm{e}(Z^s_{k,x}, g^{(r_{GID,k,x}/z_{k,x})\cdot u_{GID}}) \\ &= \mathrm{e}(g^{s\cdot z_{k,x}}, g^{(r_{GID,k,i}/z_{k,x})\cdot u_{GID}}) \\ &= \mathrm{e}(\mathrm{g}, \mathrm{g})^{s\cdot r_{GID,k,i}\cdot u_{GID}}) \end{aligned}$$

Calculate the above result

$$\begin{aligned} K &= e(N_1N_2N_3N_4) \cdot e(e(C', D^*) \cdot D) \\ &= e(g, g)^{s\cdot r_{GID}\cdot u_{GID}} \cdot e(e(g, g)^{s\cdot(y-r_{GID})} \cdot e(g, g)^{u_{GID}}) \\ &= \mathrm{e}(\mathrm{g}, \mathrm{g})^{s\cdot y\cdot u_{GID}} \\ &= Y^{s\cdot u_{GID}} \end{aligned}$$

then

$$M = C/K^{\Theta}$$

B. ESTABLISHMENT OF MUITIPLE CENTRAL AUTHORITIES

For the establishment of multiple central authorities, we have used two algorithms, which differ from those utilized for single central authorities:

1. *CASetup*: Each CA adopts this algorithm. Randomly select a random number $y_d, \mu_d \in \mathbb{Z}_p$. Assuming that $Y_d = e(g, g)^{y_d}$, where $CPK_d = Y_d$ is the public encryption parameter and $CSK_d = y_d$ is the master key of CA_d. Use the signature algorithm to derive $VerifyKey_d = \mu_d$, $CAPV_d = (SignKey_d, \mu_d)$.
2. *CAkeygen*: According to the user's GID, $D_d = e(g, g)^{o_{GID}}$, $D^* = g^{\sum y_d - r_{GID}}$, $\Theta_d = 1/o_{GID}$, and then CA_d's private identification key $SK_{CA_d} = (D_d, D^*_d)$. Next, it adopts the signature algorithm, sign $(Signkey_d, GID_d)$ to generate $\vartheta_{GID,d}$, and then obtains the generation parameter of the user's attribute key $\Xi_d = (\vartheta_{GID,d}, GID)$.

In the establishment of global master key y, it builds a polynomial threshold function with the Lagrange interpolation formula. The user can be authorized only when the number of keys exceeds the safety limit. The other algorithms are roughly the same as those of single central authorities. In a polycentric authority solution, the user accesses each central authority to obtain identity authorization information; the authorized attribute keys issued by attribute authorities are different for various central authorities.

VI. SECURITY ANALYSIS AND PERFORMANCE COMPARISON

A. SECURITY ANALYSIS

According to DBDH intractability assumption in 1.3, no Adversary $\mathcal{A}$ can distinguish the two tetrads $(g^a, g^b, g^c, (g, g)^{abc})$ and $(g^a, g^b, g^c, (g, g)^h)$ in one non-negligible polynomial time. If an Adversary can win the IND-CPA game in one non-negligible polynomial time, then it can distinguish the above mentioned two four-tuple in one non-negligible polynomial time, which are obviously contradictory.

Assume that G and G_T are two multiplication cyclic groups, of which the order is prime number p, g a generator of Group G, and then a bilinear map: e, and a, b, c, and h constitute DBDH intractability assumption, and a coin-throw set $v \in \{0, 1\}$ exists. When a challenger randomly throws a coin, if $v = 0$, then the value of Z will be $e(g, g)^{abc}$; otherwise, $e(g, g)^h$. *sim* plays a challenger and assumes that $A = g^a$, $B = g^b$, $C = g^c$, and Z, and, it simultaneously operates random coin v and takes the challenge from Adversary $\mathcal{A}$.

System Initialization: Adversary $\mathcal{A}$ initiates a request, and *sim* takes the challenge. Assume that $Y = e(A, B) = e(g, g)^{ab}$ and define the master key to be $y = a \cdot b$. Assume that $i \in \mathbb{M}$, randomly select $\alpha_i, \beta_i, \chi_i, \sigma_i \in \mathbb{Z}_p$, and then the public attribute parameter is the output. The rules are as follows:

- When attribute $i \in \mathbb{M}$ and $i = +i$, $Z_i = g^{\alpha_i}$, $Z_{n+i} = B^{\beta_i}$, $Z_{2n+i} = B^{\chi_i}$;
- When attribute $i \in \mathbb{M}$ and $i = -i$, $Z_i = B^{\alpha_i}$, $Z_{n+i} = g^{\beta_i}$, $Z_{2n+i} = B^{\chi_i}$
- When attribute $i \notin \mathbb{M}$, $Z_i = B^{\alpha_i}$, $Z_{n+i} = B^{\beta_i}$, $Z_{2n+i} = b^{\chi_i}$;
- When i is a non-leaf child node, then $t_i = b \cdot \sigma_i$ or $t_i = \sigma_i$, and the value of T_i can be evaluated.

Phase I: $\mathcal{A}$ selects an attributes set $\mathbb{I}$, which it wants to challenge, requests that attributes set $\mathbb{I} \subseteq \mathbb{M}$, and provides *sim* with identity GID, which will be used and its attributes set $\mathbb{S} \cap \mathbb{I} =$. Subsequently, *sim* receives the information from $\mathcal{A}$ and operates the key generation algorithm, and CA generates private identification key $SK_{CA} = (D, D^*, \Theta)$ and attribute key's generation parameter $\Xi_d = (CAPV_d, \vartheta_{GID,d}, GID)$; AA verifies the identity; if the identity is genuine, then the user's attribute key will be generated according to the following rules below:

Assuming that attributes set $\mathbb{S}$ does not conform to access structure $\mathbb{I}$; at least one attribute j should conform to the following conditions: $j \in \mathbb{S}$ and $\underline{j} = -j$, or $j \notin \mathbb{S}$ and $\underline{j} = +j$. Assume that everything is $j \notin \mathbb{S}$ and $\underline{j} = +j$.

Randomly select $\{r'_{GID,i}\}_{1 \leq i \leq m} \in \mathbb{Z}_p$. If $i \neq j$, $r_{GID,j} = a \cdot b + r'_{GID,j} \cdot b$, and $r_{GID,i} = r'_{GID,i} \cdot b$. Therefore,

$$r_{GID} = \sum_{k,i=1}^{k=t,i=m} r_{GID,k,i} = ab + \sum_{k,i=1}^{k=t,i=m} r'_{GID,k,i} \cdot b,$$

because

$$D^* = g^{y-r} = g^{ab-r_{GID}};$$

hence,

$$D^* = g^{\Delta} = \Pi_{k,i=1}^{k=t,i=m}(1/B^{r'_{GID,k,i}}),$$

where

$$\Delta = -\sum_{k,i=1}^{k=t,i=m} r'_{GID,k,i} \cdot b.$$

1. For the generation of the attribute key of leaf nodes $j \notin \mathbb{S}$, $j \in \mathbb{I}$, and $j = +j$ in the access tree, the rules are as follows:

- For every $i \neq j$:
When $i \in \mathbb{S}$, if $i \in \mathbb{I} \wedge \underline{i} = +i$,

$$D'_{GID,k,i} = B^{(r'_{GID,k,i}/\alpha_{k,i}) \cdot o_{GID}} = g^{(r_{GID,k,i}/\alpha_{k,i}) \cdot o_{GID}};$$

if $(i \in \mathbb{I} \wedge \underline{i} = +i) \vee i \notin \mathbb{I}$,

$$D'_{GID,k,i} = g^{(r'_{GID,k,i}/\alpha_{k,i}) \cdot o_{GID}} = g^{(r_{GID,k,i}/(b \cdot \alpha_{k,i})) \cdot o_{GID}};$$

When $i \notin \mathbb{S}$, if $i \in \mathbb{I} \wedge \underline{i} = -i$,

$$D'_{GID,k,i} = B^{(r'_{GID,k,i}/\beta_{k,i}) \cdot o_{GID}} = g^{(r_{GID,k,i}/\beta_{k,i}) \cdot o_{GID}};$$

if $(i \in \mathbb{I} \wedge \underline{i} = +i) \vee i \notin \mathbb{I}$,

$$D'_{GID,k,i} = g^{(r'_{GID,k,i}/\alpha_{k,i}) \cdot o_{GID}} = g^{(r_{GID,k,i}/(b \cdot \alpha_{k,i})) \cdot o_{GID}};$$

To get the value of F'_{GID}, initially calculate

$$\begin{aligned} F'_{GID,k,j} &= \left(A^{(1/\chi_{k,j})} \cdot g^{(r'_{GID,k,j}/\chi_{k,j})}\right)^{o_{GID}} \\ &= g^{(r_{GID,k,j}/(b \cdot \chi_{k,j})) \cdot o_{GID}}; \end{aligned}$$

If $i \in \mathbb{I}$,

$$F'_{GID,k,i} = g^{(r'_{GID,k,i}/\chi_{k,i}) \cdot o_{GID}} = g^{(r_{GID,k,i}/(b \cdot \chi_{k,i})) \cdot o_{GID}};$$

otherwise,

$$F'_{GID,k,i} = B^{(r'_{GID,k,i}/\chi_{k,i}) \cdot o_{GID}} = g^{(r_{GID,k,i}/\chi_{k,i}) \cdot o_{GID}};$$

2. To generate the attribute key of non-leaf node $x = [i \ldots j]$ in the access tree, the rules are as follows:

- For access structures in $j' \notin [i \ldots j]$,

$$F'_{GID,k,x} = \Pi_{x=i}^{j} B^{(r'_{GID,k,x}/\sigma_{k,x}) \cdot o_{GID}};$$

- For some of the access structures in $j' \in [i \ldots j]$,

$$F'_{GID,k,x} = \Pi_{x=i}^{j} g^{(r'_{GID,k,x}/\sigma_{k,x}) \cdot o_{GID}};$$

- For all access structures in $j' \in [i \ldots j]$,

$$F'_{GID,k,x} = A \cdot \Pi_{x=i}^{j} g^{(r'_{GID,k,x}/\sigma_{k,x}) \cdot o_{GID}}.$$

According to AAKeyGen, users can use the attribute key of child nodes to generate the attribute key of its parent node.

Challenging Phase: $\mathcal{A}$ selects and presents to the Challenger *sim* two messages M_0 and M_1 with the same length and an authorized access set $\mathbb{Q}$, which it wants to challenge. Challenger *sim* randomly selects $b \in \{0, 1\}$, encrypts M_b, obtains the encrypted ciphertext $CT_b = Encryption(M_b, \delta, \mathbb{Q})$, and presents the latter to the Opponent. The specific encryption process is described in Section 4

Phase II: $\mathcal{A}$ repeats the work in Phase I and continues to inquire *sim*.

Guess Phase: $\mathcal{A}$ outputs its conjecture about b, b'. If $b' = b$, then *sim* will output $v = 0$; otherwise, $v = 1$. Therefore, Adversary $\mathcal{A}$ can distinguish M_b with a non-negligible advantage ε in one polynomial time, which contradicts the DBDH intractability problem.

B. PERFORMANCE COMPARISON

If the user uses an authorized attributes set sparser then the global attributes set, then the solution in [18] can effectively reduce ciphertext length and computational expense. We have addressed these issues in this paper. This paper proposed a solution that expresses the access structure that the user develops with as few elements as possible. According to theoretical analysis, in a user-designated access structure set, if there are few concerning elements, then the solution proposed in this paper will reduce the number of indexes during encryption and parings during decryption from n to at most $\log(n)$.

In the following content, the hierarchical attribute-based encryption proposed in this paper is compared with the existing solutions. We assume that attribute elements in the number of m, l is the matrix linage in linear secret sharing scheme LSSS in the global field, L is the matrix linage which the users use during decryption; the comparative solutions are mostly multi-authority solutions, the number of CAs and AAs are not considered in this paper. Results of the comparison are as follows:

The above comparison shows that the solution proposed in this paper cannot ensure a fixed ciphertext length; however, it can effectively decrease index and paring computation during encryption and decryption, respectively, when user-designated relatively global attributes are few, which can effectively reduce internet usage and raise encryption computation efficiency in many contexts.

In order to better represent the optimization of the encryption, decryption, and cipher text lengths of the program, the following three aspects will be theoretically analyzed and proved:

First, a balanced binary tree Γ is defined, assuming that the number of leaf nodes is n, and the number of leaf nodes is 1, 2, …, n-1, n respectively. For example, Figure 2 shows. Other nodes are named after the attribute numbers contained in their child nodes respectively. . Analyze two kinds of situations: 1) When the user only sets the number 1 as the attribute that the user pays attention to (ie the attribute value is "*Positive*" or "*Negative*"), other attributes are not concerned (that is, the attribute value is "*Don*’’’ *t Care*"); 2) The user needs (or cares) for all attributes.

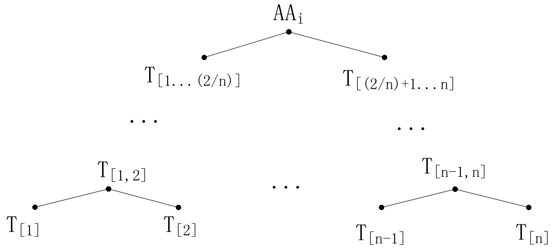


FIGURE 2. Construction of attribute tree.

For Scenario 1: According to the encryption operation, attributes 1 and 2 are child nodes of the attribute node [1], [2]. When performing the recursive operation in the algorithm, $attributes1 \in \mathbb{I}$ and $attributes2 \notin \mathbb{I}$, if (Value of attribute 1) $= +i$, $C''_{k,i} = Z^s_{k,i}$; if (Value of attribute 1) $= -i$, $C''_{k,i} = Z^s_{k,n+i}$; for (Value of attribute 2) $\in \mathbb{M} \backslash \mathbb{I}$, $C''_{k,i} = Z^s_{k,2n+i}$. Including it in ciphertext returns 1 at the same time; because other node's attribute values are "*Don't Care*," $i \notin \mathbb{I} \wedge i+1 \notin \mathbb{I}$ returns 0. Perform the operations in Encryption (2) to get the ciphertext.

Therefore, in this case, the length of the attribute ciphertext contained in the ciphertext is effectively compressed, and the length of the ciphertext is shortened. It is worth noting that the optimal effect of this scheme appears in all the attribute values "*Don 't Care*'.

For Scenario 2: $(Value\ of\ attribute) = +i$ exists for each attribute when encrypting, $(Value\ of\ attibute) = -i$ exists for $C''_{k,i} = Z^s_{k,i}$, or $C''_{k,i} = Z^s_{k,n+i}$, and it is included in ciphertext and returns 1 at the same time; the operation in Encryption(2) is finalized Ciphertext.

It can be concluded that in scenario 2, the result of computing the ciphertext is the same as the length of the attribute ciphertext contained in the ciphertext obtained by using the AND Gate under normal conditions.

In the above proof, only the number of encryption indexes and the length of the ciphertext are proved. The method of deciphering the number of pairings is similar to that of the encryption, and it is not explained here too much.

It can be seen that although the scheme of this paper is to make the length of the ciphertext constant, it can effectively reduce the redundant information in the ciphertext in the attribute-based encryption system of the "Large Universe" level. At the same time, the length of the ciphertext is reduced, the number of encryption indexes is reduced, and the number of decryption pairs is reduced.

C. THE APPLICATION OF THE SCHEME IN DRM ARCHITECTURE

As shown in Fig. 3, in the CyVOD DRM architecture, the scheme can be used for multi-platform applications for personal computers and mobile terminals (the system needs to consider terminal computing power). After the key related parameters are generated, the relevant parameters are sent

TABLE 3. Comparison of computation efficiency.

	Multi-Authority	Encryption index quantity	Decryption pairing quantity	Ciphertext size	Security	Policy
[13]	Yes	2l	2L+1	2l+2	Adaptively	LSSS
[15]	No	m+4	m<	m+6	--	Tree
[18]	Yes	c	m	m+2	Selective	AND gate
[24]	Yes	3l	3l + 1	3l + 1	Adaptively	LSSS
Our	Yes	$\log(m) \leq$	$\log(m) \leq$	$\log(m)+2 \leq$	Selective	AND gate

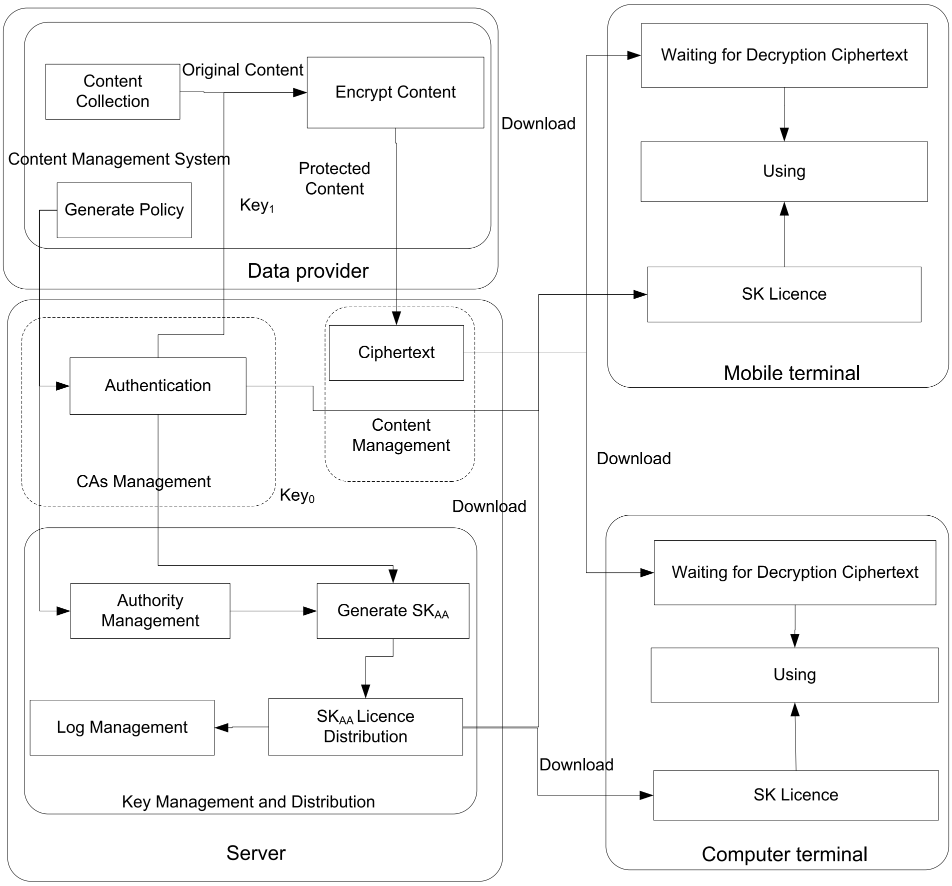


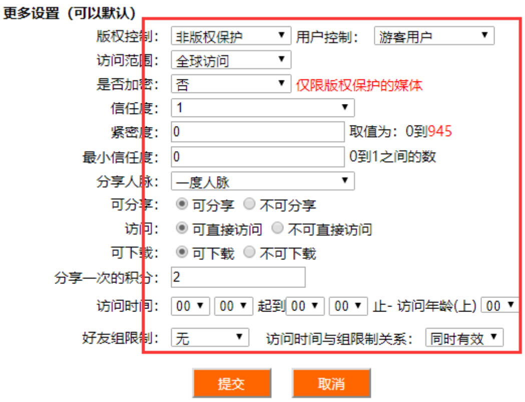
FIGURE 3. Application of scheme in CyVOD DRM architecture.

to the key management authority (attribute authorization) to generate the relevant key of the receiving user and distributed to the receiving user. When users use encrypted data, they use their own private key to decrypt (verify) that only users who meet the access policy can decrypt the data. For users who meet the access policy, they can use encrypted data on any of their own end devices. The user terminal type can be used as a kind of attribute, and the fine-grained access control to the user terminal can be realized.

The system does not need to be concerned with the way that the user obtains the encrypted data, simply verifies the relevant information of the user's identity (attribute), issues the key to the user who satisfies the access policy, and ends the authorization process. It is worth noting that the key generation service and the encrypted data storage process are simplified in the system architecture diagram.

In the application architecture diagram, the provider of the data (i.e., the data owner) is responsible for encrypting, developing access control policies, issuing ciphertext, and so on. The service provider can be divided into two categories: 1) storage service provider, 2) property key management and distribution provider; The data receiver can be used on the PC side when downloading the encrypted data, or it can be downloaded from the mobile terminal. Assume that authorized by a simple case: 1) the data provider using symmetric encryption algorithm for multimedia data or other large files encrypted (directly using attribute-based encryption is obviously not appropriate; the ciphertext produced by symmetric encryption is called “SEM”;) 2) using ABE encryption symmetric encryption key (hereinafter referred to as “AEK”), SEM and AEK are stored in storage service providers so that data recipients can download (use); 3) data receiver downloaded the SEM and AEK, at the same time it has obtained the relevant with their key attributes, AEK decrypted it to use its own attributes, thus successfully obtain symmetric encryption key to decrypt SEM.

Next, we will discuss the use of attribute-based access control policy implementation problems and context-sensitive access control methods in the system. Under the CyVOD system, development was performed using .Net technology to implement access control based on user attributes, and achieved some results [25]. The user interface is shown in Fig.3. In the system, the user only needs to input (select) the attribute feature value of the group to be authorized, and the system will sort and optimize through the user’s input and sort the user’s non-relation (ie, meaningless) attributes to obtain the minimum generation result. It is easy to see that the scale of user attributes at the current stage can be extended to an attribute authorization system with the characteristics of “Large Universe”.



更多设置（可以默认）

版权控制：非版权保护 ▾ 用户控制：游客用户 ▾

访问范围：全球访问 ▾

是否加密：否 ▾ 仅限版权保护的媒体

信任度：1 ▾

紧密度：0 取值为：0到945

最小信任度：0 0到1之间的数

分享人脉：一度人脉 ▾

可分享：● 可分享 ○ 不可分享

访问：● 可直接访问 ○ 不可直接访问

可下载：● 可下载 ○ 不可下载

分享一次的积分：2

访问时间：00 ▾ 00 ▾ 起到 00 ▾ 00 ▾ 止- 访问年龄(上) 00 ▾

好友组限制：无 ▾ 访问时间与组限制关系：同时有效 ▾

FIGURE 4. **User setting access control policy.**

In the fig.4, the part of the red box is the part of the user setting access control policy, which contains the partial default value (that is, the user does not set any value as “irrelevant” by default).

At the same time, it is necessary to add context-sensitive access control methods in the access control strategy [26]. With the development of computing and network computing, it brings new challenges to access control. Users can access the network anytime and anywhere, and the user dynamic network access environment also brings new challenges to users’ data security and privacy. In the [27], an access control design framework based on situational awareness is proposed. Reference [28] focuses on the birth of NoSQL database and discusses the fine-grained access control scheme research in this medium environment. At the same time, context-sensitive access control scheme has proposed new requirements for the attribute based encryption scheme, which has certain guiding significance in the [29].

The attributes of users often include not only the unchangeable attribute information such as gender and occupation, but also the properties of location, access time, etc. that change with time and space. So when encryption and authorization for access control strategies, the user’s private key will be empty at any time change and the request of the new key, so frequent request a new key and update will not only waste of network resources, but also reduces the safety of the ciphertext data (illegal users may to virtual space and time according to this information, so as to cheat the system). Therefore, in the next system design is based on the appearance of fine lines on the sensitive access control policy gradually with the encryption scheme based on attribute in the research and discussion, this will make the encryption scheme based on attribute is more suitable for real application scenario, to meet the needs of the user/group.

VII. CONCLUSIONS

The existing techniques have shown that the existing attribute-based encryptions based on the AND gate access structure can ensure the constant length of ciphertexts in encryption mechanisms, which support *Don’t Care* and have a constant ciphertext length. However, this kind of encryption only applies to small-scale attribute encryption systems. Meanwhile, when more *Don’t Care* values emerge in the user-designated access structure, encryption and decryption efficiency will be lowered significantly, and redundant information in the ciphertext will occur, thereby resulting in overlong ciphertexts and low internet utilization rates.

In order to address these issues, we proposed a polycentric hierarchical AND gate attribute encryption in prime order groups, which applies to system of “large universe” constructions. For the attributes administered in each AA, the element value of their child nodes is represented by the nodes of the binary tree by building an attribute binary tree. This representation can effectively decrease encryption and decryption computations and ciphertext length for the access control structure with many *Don’t Care* element values. The scheme proposed in this paper can solve the above mentioned problems effectively; however, it is limited in capacity and cannot support access control strategies with flexibility, which will be the aim of our further research. We also aim to leverage the scheme to handle the burning security issue of social media networks [30].

ACKNOWLEDGMENT

The authors would like to thank the reviewers and editor for their valuable comments, questions and suggestions.

REFERENCES

[1] A. Sahai and B. Waters, “Fuzzy identity-based encryption,” in *Proc. 24th Annu. Int. Conf. Theory Appl. Cryptograph. Techn.*, Aarhus, Denmark, Springer-Verlag, 2005, pp. 457–473.

[2] 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.*, New York, NY, USA, 2006, pp. 89–98.

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

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

[5] W. Sun, S. Yu, W. Lou, Y. T. Hou, and H. Li, "Protecting your right: Verifiable attribute-based keyword search with fine-grained owner-enforced search authorization in the cloud," *IEEE Trans. Parallel Distrib. Syst.*, vol. 27, no. 4, pp. 1187–1198, Apr. 2016.

[6] M. Chase, "Multi-authority attribute based encryption," in *Theory of Cryptography*. Berlin, Germany: Springer, 2007, pp. 515–534.

[7] J. Han, W. Susilo, Y. Mu, J. Zhou, and M. H. A. Au, "Improving privacy and security in decentralized ciphertext-policy attribute-based encryption," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 3, pp. 665–678, Mar. 2015.

[8] Q. Li, J. Ma, R. Li, J. Xiong, and X. Liu, "Provably secure unbounded multi-authority ciphertext-policy attribute-based encryption," *Secur. Commun. Netw.*, vol. 8, no. 18, pp. 4098–4109, Aug. 2015.

[9] W. Li, K. Xue, Y. Xue, and J. Hong, "TMACS: A robust and verifiable threshold multi-authority access control system in public cloud storage," *IEEE Trans. Parallel Distrib. Syst.*, vol. 27, no. 5, pp. 1484–1496, May 2015.

[10] Y. Zhang, D. Zheng, and R. H. Deng, "Security and privacy in smart health: Efficient policy-hiding attribute-based access control," *IEEE Internet Things J.*, vol. 5, no. 3, pp. 2130–2145, Jun. 2018.

[11] M. Chase and S. S. M. Chow, "Improving privacy and security in multi-authority attribute-based encryption," in *Proc. 16th ACM Conf. Comput. Commun. Secur.*, New York, NY, USA, 2009, pp. 121–130.

[12] Q. Li, J. Ma, J. Xiong, T. Zhang, and X. Liu, "Fully secure decentralized key-policy attribute-based encryption," in *Proc. 5th Int. Conf. Intell. Netw. Collaborative Syst.*, Washington, DC, USA, 2013, pp. 220–225.

[13] Z. Liu, Z. Cao, Q. Huang, D. S. Wong, and T. H. Yuen, "Fully secure multi-authority ciphertext-policy attribute-based encryption without random oracles," in *Proc. 16th Eur. Conf. Res. Comput. Secur.*, Leuven, Belgium, 2011, pp. 278–297.

[14] J. Ning, X. Dong, Z. Cao, L. Wei, and X. Lin, "White-box traceable ciphertext-policy attribute-based encryption supporting flexible attributes," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 6, pp. 1274–1288, Jun. 2015.

[15] E. Luo, Q. Liu, and G. Wang, "Hierarchical multi-authority and attribute-based encryption friend discovery scheme in mobile social networks," *IEEE Commun. Lett.*, vol. 20, no. 9, pp. 1772–1775, Sep. 2016.

[16] Z. Zhou, D. Huang, and Z. Wang, "Efficient privacy-preserving ciphertext-policy attribute based-encryption and broadcast encryption," *IEEE Trans. Comput.*, vol. 64, no. 1, pp. 126–138, Jan. 2015.

[17] 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.

[18] S. Xiao, A. Ge, and C. Ma, "Decentralized attribute-based encryption scheme with constant-size ciphertexts," *J. Comput. Res. Develop.*, vol. 53, no. 10, pp. 2207–2215, Oct. 2016.

[19] S. Wang, J. Zhou, J. K. Liu, J. Yu, J. Chen, and W. Xie, "An efficient file hierarchy attribute-based encryption scheme in cloud computing," *IEEE Trans. Inf. Forensics Security*, vol. 11, no. 6, pp. 1265–1277, Jun. 2016.

[20] N. Gorasia, R. R. Srikanth, N. Doshi, and J. Rupareliya, "Improving security in multi authority attribute based encryption with fast decryption," *Procedia Comput. Sci.*, vol. 76, pp. 632–639, Mar. 2016.

[21] J. Han, W. Susilo, Y. Mu, and J. Yan, "Privacy-preserving decentralized key-policy attribute-based encryption," *IEEE Trans. Parallel Distrib. Syst.*, vol. 23, no. 11, pp. 2150–2162, Nov. 2012.

[22] A. Beimel, "Secure schemes for secret sharing and key distribution," Fac. Comput. Sci., Technion-Israel Inst. Technol., Haifa, Israelm, Tech. Rep., 1996, pp. 76–90.

[23] D. M. Freeman, "Converting pairing-based cryptosystems from composite-order groups to prime-order groups," in *Advances in Cryptology—EUROCRYPT*. Berlin, Germany: Springer, 2011, pp. 44–61.

[24] A. Lewko and B. Waters, "Decentralizing attribute-based encryption," in *Proc. Annu. Int. Conf. Theory Appl. Cryptograph. Techn.* Berlin, Germany: Springer, 2011, pp. 568–588.

[25] Z. Zhang, L. Han, C. Li, and J. Wang, "A novel attribute-based access control model for multimedia social networks," *Neural Netw. World*, vol. 26, no. 6, pp. 543–557, Dec. 2016.

[26] A. S. M. Kayes, J. Han, and A. Colman, "OntCAAC: An ontology-based approach to context-aware access control for software services," *Comput. J.*, vol. 58, no. 11, pp. 3000–3034, Nov. 2015.

[27] C. Bauer and A. K. Dey, "Considering context in the design of intelligent systems: Current practices and suggestions for improvement," *J. Syst. Softw.*, vol. 112, pp. 26–47, Feb. 2016.

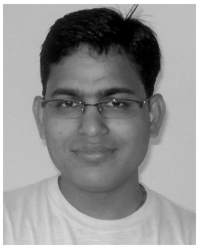
[28] P. Colombo and E. Ferrari, "Fine-grained access control within NoSQL document-oriented datastores," *Data Sci. Eng.*, vol. 1, no. 3, pp. 127–138, Sep. 2016.

[29] A. S. M. Kayes, J. Han, and A. Colman, "An ontological framework for situation-aware access control of software services," *Inf. Syst.*, vol. 53, pp. 253–277, Oct./Nov. 2016.

[30] Z. Zhang and B. B. Gupta, "Social media security and trustworthiness: Overview and new direction," *Future Gener. Comput. Syst.*, vol. 86, pp. 914–925, Sep. 2018, doi: 10.1016/j.future.2016.10.007.

ZHIYONG ZHANG (M'06–SM'11) was born in 1975. He received the master's degree in computer science from the Dalian University of Technology and the Ph.D. degree in computer science from Xidian University, China. He has been ever holding the Post-Doctoral Fellowship with the School of Management, Xi'an Jiaotong University, China. He is currently a full-time Henan Province Distinguished Professor and the Dean with the Department of Computer Science, Information Engineering College, Henan University of Science and Technology. He is also a Visiting Professor with the Computer Science Department, Iowa State University. His research interests include multimedia social networks, digital rights management, trusted computing, and usage control. Recent years, he has published over 100 scientific papers and edited six books in the above research fields, and also holds eight authorized patents. He is an ACM Senior Member of the IEEE Systems, Man, and Cybernetics Society Technical Committee on soft computing, the World Federation on Soft Computing Young Researchers Committee, and a Membership for Digital Rights Management Technical Specialist Workgroup Attached to China National Audio, Video, Multimedia System and Device Standardization Technologies Committee. He is a TPC Member for numerous international conferences/workshops on digital rights management and cloud computing security. He is a Chair or Co-Chair for numerous international conferences/workshops on digital rights management and cloud computing security. He is also an Editorial Board Member and an Associate Editor of the *Multimedia Tools and Applications* (Springer), *Neural Network World*, *EURASIP Journal on Information Security* (Springer), *Social Network Analysis and Mining* (Springer), Topic (DRM) Editor-in-Chief of the *International Journal of Digital Content Technology and its Applications*, leading a Guest Editor or a Co-Guest Editor of *Applied Soft Computing* (Elsevier), *The Computer Journal* (Oxford), and *Future Generation Computer Systems* (Elsevier).

CHENG LI was born in 1992. She is currently pursuing the master's degree in computer science with the Information Engineering College, Henan University of Science and Technology. His research interest focuses on information security, applied cryptography, and multimedia social networks security.



BRIJ B. GUPTA (GS'08–M'09–SM'17) received the Ph.D. degree in the area of information and cyber security from IIT Roorkee, India. He was also a Visiting Researcher with Yamaguchi University, Japan, in 2015. He is currently an Assistant Professor with the Department of Computer Engineering, National Institute of Technology Kurukshetra, Kurukshetra, India. In 2009, he was selected for Canadian Commonwealth Scholarship awarded by the Government of Canada. He published over 100 research papers (including two books and 14 book chapters) in International Journals and Conferences of high repute, including IEEE, Elsevier, ACM, Springer, Wiley, Taylor & Francis, and Inderscience. He has visited several countries, i.e., Canada, Japan, Malaysia, China, Hong-Kong, and so on to present his research work. His biography was selected and published in the 30th Edition of Marquis Who's Who in the World, 2012. His research interest includes information security, cyber security, cloud computing, Web security, and intrusion detection and phishing. He was a recipient of the Young Faculty Research Fellowship Award from the Ministry of Electronics and Information Technology, Government of India, in 2017. He is also working as principal investigator of various R&D projects. He is serving as an Associate Editor for the IEEE ACCESS and IJICS, Inderscience, and an Executive Editor of IJITCA, Inderscience. He is also serving as a reviewer for Journals of IEEE, Springer, Wiley, Taylor & Francis, and so on. He is also serving as a guest editor of various reputed Journals.

DANMEI NIU was born in 1979. She is currently a Lecturer with the Information Engineering College, Henan University of Science and Technology. Her research interest focuses on computer network security and usage control, and she has published above 20 papers in the above fields.