IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received December 18, 2016, accepted January 8, 2017, date of publication January 11, 2017, date of current version March 8, 2017.

Digital Object Identifier 10.1109/ACCESS.2017.2651904

A Ciphertext-Policy Attribute-Based Encryption Based on an Ordered Binary Decision Diagram

LONG LI[1], TIANLONG GU[2], LIANG CHANG[3], ZHOUBO XU[3], YINING LIU[3], AND JUNYAN QIAN[3]

[1]School of Electromechanical Engineering, Guilin University of Electronic Technology, Guilin 541004, China
[2]Guangxi Experiment Center of Information Science, Guilin University of Electronic Technology, Guilin 541004, China
[3]Guangxi Key Laboratory of Trusted Software, Guilin University of Electronic Technology, Guilin 541004, China

Corresponding author: L. Chang (changl@guet.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61572146, Grant 61363030, Grant 61262030, Grant U1501252, and Grant 61562015, in part by the Natural Science Foundation of Guangxi Province under Grant 2015GXNSFAA139285, Grant 2014GXNSFAA118354, Grant 2016GXNSFDA380006, and in part by the High Level of Innovation Team of Colleges and Universities in Guangxi and Outstanding Scholars Program.

ABSTRACT Ciphertext-policy attribute-based encryption (CP-ABE) is widely used in many cyber physical systems and the Internet of Things for guaranteeing information security. In order to improve the performance and efficiency of CP-ABE, this paper makes a change to the access structure of describing access polices in CP-ABE, and presents a new CP-ABE system based on the ordered binary decision diagram (OBDD). The new system makes full use of both the powerful description ability and the high calculating efficiency of OBDD. First, in the access structure, the new system allows multiple occurrences of the same attribute in a strategy, supports both positive attribute and negative attribute in the description of access polices, and can describe free-form access polices by using Boolean operations. Second, in the key generation stage, the size of secret keys generated by the new system is constant and not affected by the number of attributes; furthermore, time complexity of the key generation algorithm is O(1). Third, in the encryption stage, both the time complexity of the encryption algorithm and the size of generated ciphertext are determined by the number of valid paths contained in the OBDD instead of the number of attributes occurring in access polices. Finally, in the decryption stage, the new system supports fast decryption and the time complexity of the decryption algorithm is only O(1). As a result, compared with existing CP-ABE schemes, the new system has better performance and efficiency. It is proved that the new CP-ABE system can also resist collision attack and chosen-plaintext attack under the decisional bilinear Diffie Hellman assumption.

INDEX TERMS Ciphertext-policy attribute-based encryption, ordered binary decision diagram, access structure, access policy, decryption.

I. INTRODUCTION

In certain network scenarios, such as the Internet of things (IoT) and cyber physical systems (CPS), users and nodes of diverse types are located in different geographic regions. The relationships between these entities are complicated; a data owner often needs to maintain a one-to-many relationship and provide services to more than one unknown user. Secure information transmission and effective access control is challenging.

To guarantee the security of information to be shared and prevent unauthorized access, a simple and straightforward approach is to encrypt the data beforehand. The most sophisticated encryption method is public key encryption, which is widely used. Traditional public key encryption requires two types of keys: a public key to encrypt the plaintext and a private key to decrypt the ciphertext. Since there are many users of systems such as the IoT, the overhead in encryption, key generation, management and maintenance will be prohibitively large if traditional public key encryption is used to encrypt and decrypt messages. Besides, in scenarios such as the IoT and CPS, the exact identities and number of users cannot be acquired beforehand, further impeding the implementation of traditional public key encryption. These limitations create favorable conditions for attribute-based encryption (ABE).

ABE was first proposed by Sahai and Waters [1] on the basis of identity-based encryption (IBE) and in subsequent research was expanded into two different branches:

ciphertext-policy ABE [2] (CP-ABE) and key-policy ABE [3] (KP-ABE). In CP-ABE, the encryptor processes encryption with the help of an access structure (i.e., the public key), and decryption is successful if and only if the decryptor's attribute set (i.e., the private key) meets the access structure requirements. For example, using such kind of algorithms, cloud servers in the IoT can specify the following terms for data interaction with certain terminals: (''deployment time > 3 years'' AND ''located in outdoor'' AND (''equipped with humidity sensors'' OR ''equipped with temperature sensors'')). Since CP-ABE employs only descriptive attributes instead of exact identities and numbers of users, CP-ABE is well suited for encryption and decryption of shared messages in large-scale networks. This capability resolves the above problems and is applicable to scenarios such as the IoT.

Some components of CP-ABE still need to be improved or optimized, such as the representation of access policies, the efficiency of encryption and decryption. As the basis of CP-ABE schemes, access structure plays an important role in the design of CP-ABE. Several types of access structure have been proposed to represent access policies, including threshold gates [2], the LSSS matrix [4], AND gates [5] and the distribution matrix [6] and so on. All of these access structures can successfully represent access policies, but efficiency and flexibility are not ideal. Certain other drawbacks still exist in the above access structures; for example, the secret to be shared must be an integer [4]. Therefore, the study of access structures provides an opportunity to improve efficiency and flexibility of CP-ABE schemes.

An ordered binary decision diagram (OBDD) is a structure that can not only realize the representation of Boolean functions [7] but also efficiently implement operations between Boolean variables and Boolean functions [8]; this special structure is an ideal choice to describe the access policies of CP-ABE.

Based on OBDD, this paper proposes a non-monotonic, expressive and flexible access structure. This structure supports both positive attributes and negative attributes without increasing system overhead; it also supports multiple occurrences of attributes and all Boolean operations such as AND, OR and NOT between attributes. Furthermore, a new CP-ABE scheme is proposed based on the above access structure, which offers better performance in terms of encryption, key generation and decryption, resists collusion attacks and is CPA secure under the decisional bilinear Diffie-Hellman (DBDH) assumption. *To the best of our knowledge, this is the first attempt to introduce the concept of OBDD into the design of ABE.*

The rest of this paper is organized as follows. Related work is summarized in Section 2. Background knowledge related to OBDD and CP-ABE is introduced in Section 3. The detailed design of the OBDD access structure, the main construction, security proof and performance analysis of the new CP-ABE are described in Section 4. Conclusions and recommendations for future work are provided in section 5.

II. RELATED WORK

The design of CP-ABE was first proposed by Bethencourt et al. [2]; in this approach, the encryption algorithm encrypts a message under an access tree, and the decryption algorithm decrypts encrypted messages using Lagrange interpolation.

CP-ABE has received considerable attention since it was proposed. In recent years, both access structures and security proofs have become active areas of research, and a number of research results of theoretical significance and/or practical value have been published.

Waters [4] proposed a flexible access structure based on LSSS, designed a CP-ABE construction method and further constructed three different CP-ABE schemes based on several intractability assumptions. All of these schemes improved certain aspects, such as ciphertext size and private key size, but a drawback in the construction method is that each attribute can occur only once in an access structure. Although the paper proposes a solution to this problem, the solution degrades the performance.

In the scheme proposed by Ling and Newport [5] of MIT CSAIL, a new structure composed of AND gates is used. The scheme is CPA secure under DBDH assumption and supports both positive and negative attributes. To improve efficiency and increase security, hierarchical attributes and the Canetti-Halevi-Katz technique are applied, and a chosen-ciphertext (CCA) secure extension is obtained. It should be noted that this is the first formal CCA security proof for CP-ABE.

By employing a secret sharing technology termed LISS and describing the access policy in a distribution matrix, Balu et al. [6] constructed a CP-ABE scheme that supports multiple occurrences of attributes. The disadvantage is that the message to be encrypted must be an integer within the pre-defined range $[-2^l, 2^l]$.

Addressing the problem of key update, Rao and Dutta [9] proposed a CP-ABE that can perform attribute level dynamic operations at an ideal cost. However, the access structure used to describe the access policy is still a traditional monotone tree composed of threshold gates.

From the perspective of privacy preserving, Zhou et al. [10] proposed privacy preserving constant CP-ABE, which can compress the ciphertext to a constant size while ignoring the number of attributes contained in the system. This scheme offers favorable performance, but it cannot support access policies in non-monotonic form or disjunctive normal form.

Substantial research on different pointcuts has been conducted. The problem of key management is studied in [11]; based on a hierarchical organization of users, a CP-ABE scheme that can realize delegation of access rights is presented. The concept of user groups is adopted by [12] to solve the permission revocation problem, and outsourcing computation is used to improve efficiency and performance. By combining techniques of ABE with proxy re-encryption and lazy re-encryption, Yu et al. [13] proposed a scheme that allows the data owner to delegate most of the computation

tasks to untrusted cloud servers without any information disclosing. By proposing a multi-authority ABE, Chase and Chow [14] resolved the key-escrow problem. To trace the identity of a misbehaving user who leaked the decryption key to others, Li et al. [15] proposed a multi-authority CP-ABE scheme with accountability.

At the application research of the thesis, lots of works have been done. ABE is used by [16]–[18] to protect the security of personal health record stored in cloud server and other details such as scalability and dynamics. Smari et al. [19] present a more general and flexible access structure and an extended access control model for high performance distributed collaboration environments. To guarantee the security and privacy in the friend discovery process of mobile social networks, Luo et al. [20] proposed a hierarchical multi- authority and ABE friend discovery scheme based on CP-ABE.

III. BACKGROUND KNOWLEDGE

In this section, background knowledge related to CP-ABE and OBDD is introduced, mainly including important concepts, special syntaxes and basic algorithms.

A. ACCESS STRUCTURE

The essence of an access policy is a rule R that returns 1 or 0 according to the state of an attribute set S. The rule R will return 1 if and only if S satisfies R, written as $S \vDash R$, and 0 is returned by R when S does not satisfy R, written as $S \nvDash R$.

Access structures are intuitive expressions of access policies and have several mathematical forms, such as Boolean expressions and threshold gates. Threshold gates [2], [3], [6], [11], [12] are the most common form of access structures, by which the access policies can be described as element matches between two attribute sets. AND gates [5], [10], [16] are another frequently used form of access structure.

B. CP-ABE FRAMEWORK

A common framework of CP-ABE contains four algorithms:

Setup, the algorithm is executed by the authority in charge of the generation of the public key PK and master key MK.

Encrypt, the algorithm is executed by the data owner to encrypt plaintext M.

Keygen, the algorithm is executed by the authority and generates a secret key SK according to the attribute set S provided by a user.

Decrypt, the algorithm is executed by the data user to decrypt a ciphertext CT with a pre-generated secret key.

C. CPA SECURITY GAME FOR CP-ABE

Definition 1 (CPA Security of CP-ABE Scheme): If there is no probabilistic polynomial time within which adversaries can win the following game with non-negligible advantage, the CP-ABE scheme is said to be secure against the chosen plaintext attacks.

Initial, the adversary chooses a specific access structure AS and submits it to the challenger.

Setup, the challenger runs the ***Setup*** algorithm in CP-ABE and submits the freshly generated PK to the adversary.

Phase1, the adversary makes a secret key query to the ***Keygen*** algorithm in CP-ABE using the attribute set S, but the restriction $S \nvDash AS$ must hold. This procedure can be repeated adaptively.

Challenge, the adversary submits two plaintexts M_0 and M_1 of equal length to the challenger. After receiving these two messages, the challenger chooses $\mu \in \{0, 1\}$ randomly and encrypts M_μ under AS to obtain the ciphertext CT. Finally, CT is passed to the adversary.

Phase2, same as ***Phase1***.

Guess, the adversary guesses the value of μ as μ'.

In the above CPA security game, the advantage of adversary A is defined as follows: $Adv^{\mathrm{CPA}}_{\text{CP-ABE}}(\mathsf{A}) = \left| Pr\left[\mu = \mu'\right] - \frac{1}{2} \right|$.

D. BILINEAR MAPS AND BILINEAR GROUPS

Definition 2 (Bilinear Maps): Let G_0 and G_1 be two multiplicative cyclic groups of prime order p. Let g be a generator of G_0 and e be a bilinear map e: $G_0 \times G_0 \rightarrow G_1$. The bilinear map e has the following properties: ① Bilinearity: for all u, $v \in G_0$ and $a, b \in \mathbb{Z}_p$, we have $e(u^a, v^b) = e(u, v)^{ab}$. ② Non-degeneracy: $e(g, g) \neq 1$.

Definition 3 (Bilinear Groups): The group G_0 is defined to be a bilinear map if the group operation in G_0 and the bilinear map e are both efficiently computable.

E. DBDH ASSUMPTION

Let e be a bilinear map e: $G_0 \times G_0 \rightarrow G_1$. Let a, b, c, z be chosen randomly from $\mathbb{Z}_p$ and g be a generator of G_0. The DBDH assumption can be described as follows: no probabilistic polynomial-time adversary can distinguish the tuple $(g^a, g^b, g^c, e(g, g)^{abc})$ from the tuple $(g^a, g^b, g^c, e(g, g)^z)$.

For any probabilistic polynomial-time adversary A, the advantage in solving the DBDH problem is defined as follows:

$$Adv^{G_0}_{\mathrm{DBDH}}(\mathsf{A}) = |Pr\left[\mathsf{A}(g, g^a, g^b, g^c, e(g, g)^{abc}) = 1\right] - Pr\left[\mathsf{A}(g, g^a, g^b, g^c, e(g, g)^z) = 1\right]|.$$

The DBDH assumption holds if $Adv^{G_0}_{\mathrm{DBDH}}(\mathsf{A})$ is negligible.

F. OBDD

A binary decision diagram (BDD) is essentially a data structure that can be used to conduct representation and manipulation of Boolean functions. A BDD can be transferred into an OBDD by specifying the variable ordering. The study of BDDs and OBDDs can be traced to [7], [8], and [14].

Definition 4 (BDD): As a special expression of the Boolean function $f(x_0, x_1, \ldots, x_{n-1})$, a BDD over a set of Boolean variables $\{x_0, x_1, \ldots, x_{n-1}\}$ and a terminal set $\{0, 1\}$ is a directed acyclic graph with exactly one root node and the following properties:

① A node in BDD is either non-terminal or terminal.

② Each non-terminal node u can be described as a tuple (f^u, var, *low*, *high*). The corresponding Boolean function of u is f^u (for root node, $f^u = f(x_0, x_1, \ldots, x_{n-1})$), the variable embedded in u is var, and the two child nodes of u are *low* (when u.var=0) and *high* (when u.var=1).

③ Each terminal node i is labeled with a constant from {0, 1} and has no child node.

④ Each non-terminal node u has two edge points to *low* and *high* separately; the edge point to *low* (*high*) is called the 0-branch (1-branch).

⑤ Each variable appears at most once on any directed path from the root node to a terminal node.

In the graphical representation, the terminal nodes (non-terminal nodes) are represented by boxes (circles), and the 0-branch (1-branch) is represented by dotted lines (solid lines).

Definition 5 (OBDD): In a BDD representation of the Boolean function $f(x_0, x_1, \ldots, x_{n-1})$, if the ordering of variables is fixed as $\boldsymbol{\pi}$, the BDD is more accurately called an OBDD.

In an OBDD, all of the variables occurring on any directed path from the root node to a terminal node are encountered in the same order $\boldsymbol{\pi}$.

IV. AN OBDD-BASED CP-ABE SCHEME

The following will be discussed in this section: OBDD access structure, the main construction, security proof and efficiency analysis of the new CP-ABE scheme designed on the basis of OBDD.

A. OBDD-BASED ACCESS STRUCTURE

A flexible and efficient OBDD access structure is proposed. This special access structure is non-monotonic, meaning it can support both positive and negative attributes; additionally, repeated appearance of attributes and Boolean operations between attributes are supported [7].

After describing an access policy in natural language, the process of generating the corresponding OBDD access structure is as follows:

1) OBTAIN THE BOOLEAN FUNCTION OF AN ACCESS POLICY

We assume that all of the attributes appearing in the access policy are numbered as i ($0 \le i \le n-1$) in a pre-defined sequence and represented by $x_i (0 \le i \le n-1)$, in which n is the total number of attributes. Then, the access policy can be converted into a Boolean function $f(x_0, x_1, \ldots, x_{n-1})$.

All Boolean functions can be transferred to basic logical operations between variables (i.e., AND, OR and NOT), but as a special type of Boolean operation, the transformation of a threshold gate is more complicated.

Definition 6 (Threshold Gates): For a threshold operation that involves n attributes if and only if users that own arbitrary t attributes (a subset of the above n attributes) can finish the operation successfully, then the operation is called a threshold gate, written as $T(t, n)$.

In some security systems, only users who can successfully finish certain specified threshold operations own access permissions to the system or have the ability of decryption.

Assuming n attributes form an attribute set N, then the Boolean function of a given threshold gate $T(t, n)$ is constructed by the following steps:

Step 1: Choose all subsets of N that contain t different attributes. According to the formulas for permutation and combination, compute the total number of such subsets $C(n, t)$. These subsets can be separately denoted by $Com_1, Com_2, \ldots, Com_{C(n,t)}$.

Step 2: For each subset with size t, a set-level conjunctive operation is performed. In other words, each conjunctive formula contains t different attributes, and there are $C(n, t)$ such formulas denoted separately by $Con_1, Con_2, \ldots, Con_{C(n,t)}$.

Step 3: The final Boolean function of $T(t, n)$ is obtained by a disjunctive operation on the above $C(n, t)$ conjunctive formulas. The final Boolean function is $f(t, n) = \bigvee_{i=1}^{C(n,t)} Con_i$.

2) OBTAIN THE OBDD-BASED ACCESS STRUCTURE

The construction of *OBDDs* used to represent Boolean functions is completed by applying a simple recursive process. As Shannon's expansion theorem shows, $f(x_0, x_1, \ldots, x_{n-1}) = x_i \cdot f_{|x_i=1} + x_i' \cdot f_{|x_i=0}$ ($0 \le i \le n-1$); thus, the recursive algorithm with pre-defined variable ordering $\boldsymbol{\pi}$: $x_0 < x_1 < \ldots < x_{n-1}$ can be described as follows:

For the same Boolean function, different variable orderings will result in different *OBDDs*; specifically, the number of nodes contained in each *OBDD* may vary, or the storage occupied by each *OBDD* may vary. Therefore, to ensure a unique *OBDD*, variable ordering $\boldsymbol{\pi}$ must be specified before the construction of the *OBDD*. The *node* in the above program is the structure used to represent the nodes of the *OBDD*. The *Computed table* is a dictionary, which is used to store already computed results of previous *Construct* - calls.

After the construction, all of the nodes contained in *OBDD* should be numbered to obtain the final expression: $OBDD = \{Node_{id}^{i} | id \in ID, i \in I\}$, in which *ID* is a set that contains all of the serial numbers of non-terminal nodes and I is a set formed by all of the attributes appearing in the access structure. $Node_{id}^{i}$ is a tuple $< id, i, high, low >$, in which *id* is the serial number of *current node*, i is the serial number of the attribute contained in *current node*, *high* is the serial number of the 1-branch node, and *low* is the serial number of the 0-branch node. The parameters *high* and *low* are used to maintain the relationships between parent nodes and child nodes. The nodes with serial numbers 0 (i.e., $\boxed{0}$) and 1 (i.e., $\boxed{1}$) have fixed meanings and the i, *high* and *low* domains of these two special nodes are meaningless, so these nodes are deleted in OBDD-based access structures to reduce the storage cost.

Definition 7 (Satisfying an OBDD): Let *OBDD* be an access structure and S be an attributes set. Starting from

the *root*, a comparison based on the values of attributes is made as follows: for a non-terminal node with attribute i, if the attribute valued 1 is contained in S, the comparison will be delivered to the 1-branch node; otherwise, the comparison must be delivered to the 0-branch node. The above process is executed repeatedly until one of the terminal nodes is reached. If the terminal node $\boxed{1}$ is finally reached, the set S satisfies the *OBDD*, denoted by $S \vDash OBDD$. On the contrary, the set S does not satisfy the *OBDD*, denoted by $S \nvDash OBDD$.

The following example intuitively explains how to construct an *OBDD* with a Boolean function that is transferred from access policy 1.

Example 1: Converting an access policy into an OBDD- based access structure.

Access policy 1: Users who own attribute x_0 or any two attributes among $\{x_1, x_2, x_3\}$ can finish the decryption successfully.

Access policy 1 contains a threshold gate T(2, 3). According to access policy 1 and the above method used to obtain the Boolean function of an access policy, the corresponding Boolean function obtained is $f_1(x_0, x_1, x_2, x_3) = x_0 + x_1x_2 + x_1x_3 + x_2x_3$.

Let the variable ordering be π: $x_0 < x_1 < x_2 < x_3$; then, the *OBDD* of the Boolean function $f_1(x_0, x_1, x_2, x_3) = x_0 + x_1x_2 + x_1x_3 + x_2x_3$ is constructed as follows: according to Shannon's expansion theorem $f(x_0, x_1, \ldots, x_{n-1}) = x_i \cdot f_{|x_i=1} + x_i' \cdot f_{|x_i=0}$ $(0 \le i \le n-1)$, the *OBDD* representation of this Boolean function is constructed using a recursive procedure (see ALGORITHM 1).

Secondly, all of the nodes should be renumbered from top to bottom and left to right to obtain the final expression as $OBDD = \{Node_{id}^{i} | id \in ID, i \in I\}$.

The access structure is finally described as $OBDD = \{\text{Node}_2^0, \text{Node}_3^1, \text{Node}_4^1, \text{Node}_5^2, \text{Node}_6^2, \text{Node}_7^2, \text{Node}_8^3, \text{No de}_9^3, \text{Node}_{10}^3\}$.

Definition 8 (A Valid Path): In *OBDD*, each path derived from *root* and ended at terminal node $\boxed{1}$ is called a valid path, denoted by $root \rightarrow \boxed{1}$.

For example, in Fig. 1, the path $(x_0 \rightarrow x_1 \rightarrow x_2' \rightarrow x_3' \rightarrow \boxed{1})$ is a valid path, but the path $(x_0' \rightarrow x_1' \rightarrow x_2' \rightarrow x_3 \rightarrow \boxed{0})$ is not a valid path.

B. MAIN CONSTRUCTION OF OBDD-BASED CP-ABE

The CP-ABE scheme proposed in this paper supports both positive attribute i and negative attribute $\neg i$, therefore, to obtain a terse statement, the symbol $\underline{i}$ (i or $\neg i$) introduced in [5] is used to represent the attribute.

Assuming the attribute set N contains n elements with serial number $\{0, 1, \ldots, n-1\}$, the CP-ABE design based on the OBDD structure consists of the following four basic algorithms.

Setup, this algorithm is executed by the authority and finishes the following operations:

ALGORITHM 1 Obtain the *OBDD* corresponding to a Boolean function

Inputs: A Boolean function f and the maximum index of variables n-1

Output: The *OBDD* representation of f with the variable ordering $\pi: x_0 < x_1 < \ldots < x_{n-1}$

```
(1)   # define max n-1
(2)   node* Construct-step(char *f, int i);
(3)   node* Construct(char *f) {
(4)     int i = 0;
(5)     node *u;
(6)     Empty the computed table;
(7)     return (u = Construct-step(f, i));
(8)   }
(9)   node* Construct-step(char *f, int i) {
(10)    static int id=1;
(11)    node*u, *v0, *v1;
(12)    if (i>max) {
(13)     if (*f == "0") u->id = 0;
(14)     else u->id = 1;
(15)     return u;
(16)    }
(17)    else {
(18)    v0=Construct-step(f|xi=0, i+1);
(19)    v1=Construct-step(f|xi=1|, i+1);
(20)    if computed-table entry (v0, v1, u) exists return u;
(21)     u->index = i;
(22)     u->id = ++id;
(23)     u->low = v0;
(24)     u->high = v1;
(25)     Store (v0, v1,u) in computed table;
(26)     return u;
(27)    }
(28)  }
```

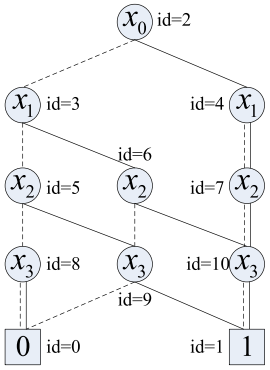


FIGURE 1. OBDD representation of $f_1(x_0, x_1, x_2, x_3)$.

Select a bilinear group G_0 of primer order p and a random generator g, with bilinear maps e: $G_0 \times G_0 \rightarrow G_1$. Randomly select $y, t_0, t_1, \ldots, t_{n-1}, t_0', t_1', \ldots, t_{n-1}'$ in $\mathbb{Z}_p$.

Define $Y := e(g, g)^y$, $T_i := g^{t_i} (i \in N)$, $T_i' := g^{t_i'} (i \in N)$, then generate the public key $PK := < e, g, Y, (T_i, T_i') | i \in N >$ and master secret key $MK := < y, (t_i, t_i') | i \in N >$.

T_i (t_i) and T_i' (t_i') correspond to the positive value and negative value of attribute i, respectively.

Encrypt $(PK, M, OBDD)$, this algorithm is executed by the data owner to encrypt plaintext M. The plaintext owned by encryptor is $M \in G_1$, and the access structure is $OBDD = \{Node_{id}^{i} | id \in ID, i \in I\}$. Assuming the number of valid paths ($root \rightarrow \boxed{1}$) is T, which can be expressed as $R = \{R_0, R_1, \ldots, R_{T-1}\}$. Encryption operations are performed as follows:

Randomly select $s \in \mathbb{Z}_p$ and compute $\tilde{C} := M \cdot Y^s$, $\hat{C} := g^s$. The ciphertext element C_{Rt} ($R_t \in R$) related to path R_t is described as $C_{R_t} := \left(\prod_{i \in I} \underline{T_i}\right)^s = g^{(\sum_{i \in I} \underline{t_i} \cdot s)}$. Since $\underline{T_i}$ corresponds to the value of attribute $\underline{i}$, $\sum \underline{t_i}$ is the information related to all attributes included in R_t; by using this parameter, all of the attributes contained in path R_t are bound together. The corresponding relationship between (T_i, T_i') and the value of $\underline{i}$ can be illustrated by the following figure (see Fig. 2).

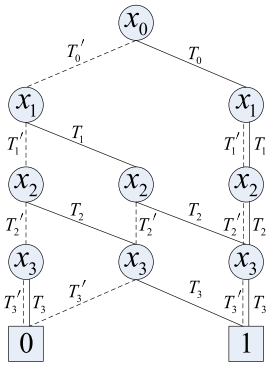


FIGURE 2. Relationship between (T_i, T_i') and the value of i.

The ciphertext obtained is $CT :=< OBDD, \tilde{C}, \hat{C}, \{C_{R_t} | R_t \in R\} >$.

In the above encryption algorithm, the main calculations include ($|I$-1$|\cdot T$) multiplication in G_0, T+1 exponential operations in G_0, one exponential operation and one multiplication in G_1. The main storage cost of ciphertext CT includes the $OBDD$ structure, one element in G_1 and T+1 elements in G_0.

Keygen (S, MK), this algorithm is executed by the authority and generates a secret key SK according to the attribute set S provided by a user. The attribute set owned by a user is denoted by S; for any attribute $i \notin S$, the default value is defined as $\neg i$.

The ***Keygen*** algorithm runs as follows:

For $i \in I$, define the value of $\underline{t_i}$: if $i \in S \wedge \underline{i} = i$, let $\underline{t_i} = t_i$; otherwise, let $\underline{t_i} = t_i'$. Select $r \in \mathbb{Z}_p$ randomly and compute $\hat{D} := g^{y-r}$, $D := g^{(r/\sum_{i \in I} \underline{t_i})}$. The generated secret key is $SK :=< \hat{D}, D >$.

The secret keys generated by this algorithm associate with the attribute set I. This measure is reasonable and effective and not only significantly reduces the computational and storage cost of authority (especially when I occupies a small part of N) but also avoids certain operations such as re-encryption and key re-generation originating from the change of global attributes (N changes when I remains unchanged).

If the number of encryptor is large and the access structures adopted are different, a measure that extends the attribute set I into the global attribute set N can be taken in advance.

Decrypt (CT, SK), this algorithm is executed by the data user to decrypt a ciphertext CT with a pre-generated private key SK. Assuming the ciphertext needed to be decrypted is $CT :=< OBDD, \tilde{C}, \hat{C}, \{C_{R_t} | R_t \in R\} >$ and the private key owned by the data user is $SK :=< \hat{D}, D >$, then the decryption process can be implemented by the following recursive algorithm:

① Seek the node with serial number 2 (i.e., *root*) and define it as the *current node*.

② Extract the information $Node_{id}^{i}$ contained in the *current node*, for attribute i: if $i \in S \wedge \underline{i} = i$, go to ③; otherwise, if $i \in S \wedge \underline{i} = \neg i \vee i \notin S$, go to ③.

③ Search the 1-branch node of the *current node* according to the *high*.

a) If the 1-branch node is the terminal node $\boxed{0}$, terminate the recursive algorithm and return decryption failure.
b) If the 1-branch node is the terminal node $\boxed{1}$, go to ⑤.
c) If the 1-branch node is a non-terminal node, define it as the *current node* and go to ②.

④ Search the 0-branch node of the *current node* according to the *low*.

a) If the 0-branch node is the terminal node $\boxed{0}$, terminate the recursive algorithm and return decryption failure.
b) If the 0-branch node is the terminal node $\boxed{1}$, go to ⑤.
c) If the 0-branch node is a non-terminal node, define it as the *current node* and go to ②.

⑤ Store the special $root \rightarrow \boxed{1}$ path R_t. Complete the following computations sequentially:

$$e\left(\hat{C}, \hat{D}\right) \cdot e\left(C_{R_t}, D\right) = e\left(g, g\right)^{s \cdot (y-r)} \cdot e\left(g, g\right)^{s \cdot r} = e\left(g, g\right)^{s \cdot y} = Y^s.$$

The plaintext can be recovered based on the formula $M = \tilde{C}/Y^s = \tilde{C}/e(g,g)^{s \cdot y}$. Then, terminate the recursive algorithm and return decryption success. □

The above derivation shows that the maximum calculation of the ***Decrypt*** algorithm includes two pairings and two multiplications in G_1, which occurs only when $SK \vDash OBDD$.

Suppose a user owns an attribute set $S = \{x_1, x_3\}$ that satisfies the OBDD-based access structure generated in the above example (see Fig. 2). The decryption path and corresponding encryption elements are shown in the figure below, which means this user is able to complete the decryption and obtain the plaintext.

C. PROOF OF CPA SECURITY

Since random number$r$ are used to generate a private key in the ***Keygen*** algorithm and all of the attributes in set I are bound by D, the CP-ABE scheme proposed in this paper can resist collision attack.

TABLE 1. Capacities analysis and comparision.

Scheme \ Indicator	Access Structure	Operations supported				Variables number	Nodes number
		AND	OR	Threshold	NOT		
LN [5]	AND gates	√	×	×	×	\	\
BSW [2], RD [9]	Threshold gates	√	√	√	×	6	10
Our scheme	OBDD	√	√	√	√	3	6

The rest of this section will prove the security of the CP-ABE scheme by reducing the CPA security to the DBDH assumption.

Theorem 1. A simulator Sim can be constructed to solve the DBDH problem with non-negligible advantage if a probabilistic polynomial-time adversary Adv can win the CP-ABE game with non-negligible advantage.

Proof. Suppose adversary *Adv* can win the CP-ABE game with advantage ε. A simulator *Sim* will be constructed to solve the DBDH problem with advantage $\varepsilon/2$.

Let G_0 be a group with prime order p and assume that a bilinear maps e: $G_0 \times G_0 \rightarrow G_1$. The challenger randomly selects 6 elements from different domains: a, b, c, $z \in \mathbb{Z}_p$, $v \in \{0, 1\}$ and a generator $g \in G_0$, then defines Z based on the value of v. If $v = 0$, $Z = e(g, g)^{abc}$; otherwise, $Z = e(g, g)^z$. Finally, challenger passes the tuple $< g, A, B, C, Z >=< g, g^a, g^b, g^c, Z >$ to the simulator *Sim*, and *Sim* will play the role of the challenger in the following process.

Init. *Adv* passes the access structure $OBDD = \{Node_{id}^i | id \in ID, i \in I\}$ to *Sim*.

Setup.*Sim* defines $Y = e(A, B) = e(g, g)^{ab}$ and selects $(t_i, t_i') \in \mathbb{Z}_p$ for $i \in I$.

Phase 1. *Adv* submits an attribute set S in the secret key query, where $S \nvDash OBDD$, which means S cannot satisfy any valid path of *OBDD*. In other words, at any valid path, there must exist an attribute $j \in I$ that satisfies either $j \in S \wedge \underline{j} = \neg j$ or $j \notin S \wedge \underline{j} = j$. Without loss of generality, *Sim* chooses an attribute that satisfies $j \notin S \wedge \underline{j} = j$.

The components related to each attributes are assigned as follows: for $j \notin S \wedge \underline{j} = j$, $\underline{t_j} = b \cdot t_j'$; for $i \neq j$, several cases are contained:

1) $i \in S \wedge \underline{i} = i$, $\underline{t_i} = t_i$;
2) $i \in S \wedge \underline{i} = \neg i$, $\underline{t_i} = b \cdot t_i$;
3) $i \notin S \wedge \underline{i} = \neg i$, $\underline{t_i} = t_i'$;
4) $i \notin S \wedge \underline{i} = i$, $\underline{t_i} = b \cdot t_i'$.

The components of the secret key are computed as follows: $\hat{D} := g^{ab-r}$, $D := g^{(r/\sum_{i \in I} \underline{t_i})}$.

Challenge. *Adv* submits plaintext M_0 and M_1 of equal length. *Sim* randomly selects $\mu \in \{0, 1\}$ and defines $\tilde{C} = M_{\underline{\mu}} \cdot Z$. The ciphertext $CT :=< OBDD, \tilde{C}, C, \{C_{R_t} = g^{\sum_{i \in I} \underline{t_i} \cdot c} | R_t \in R >$ is generated and passed to *Adv*.

Phase 2. Same as ***Phase 1.***

Guess.*Adv* provides a guess μ' of μ. If $\mu = \mu'$, *Sim* outputs "DBDH"; otherwise, it outputs "Random".

If $Z = e(g, g)^{abc}$, *CT* is a valid ciphertext, and in this case the advantage of *Adv* in winning the game is ε.

$$P[Sim \rightarrow \text{"DBDH"} | Z = e(g, g)^{abc}] = P[\mu = \mu' | Z = e(g, g)^{abc}] = 1/2 + \varepsilon.$$

If $Z = e(g, g)^z$, the ciphertext $M_\mu \cdot Z$ is absolutely random; therefore, the probability of $\mu \neq \mu'$ is exactly 1/2.

$P[Sim \rightarrow$ "Random"$|Z = e(g, g)^z] = P[\mu = \mu' | Z = e(g, g)^z] = 1/2$.

Based on the above analysis, the advantage of *Sim* in solving the DBDH problem is $(1/2 * (1/2 + \varepsilon) + 1/2 * 1/2) - 1/2 = \varepsilon/2$.

D. ANALYSIS OF CAPACITIES AND EFFICIENCY

Our scheme supports both positive attributes and negative attributes in the description of access polices without increasing system overhead; besides, our scheme supports the multiple occurrence of an attribute in the same strategy, and can describe free-form access polices by making use of any Boolean operation. All of the above features lead to a more powerful and more efficient scheme.

To explain the capacities and efficiency of our scheme in the representation of access policies, the most frequently used access structures, threshold gates [2], [9] and AND gates [5] will be analyzed along with our scheme in the following example.

Example 2: Capacities and efficiency in the representation of access policies.

Assuming an access policy is described by Boolean function: $f_2(x_0, x_1, x_2) = x_0 x_1 + x_0' x_2 + x_1' x_2'$. Two different types of graphical representations (i.e., access structures) of $f_2(x_0, x_1, x_2)$ are shown in Fig. 3, and their performances are analyzed and compared in Table 1.

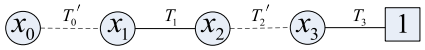


FIGURE 3. Decryption path and encryption elements of $S = \{x_1, x_3\}$.

The AND gates do not possess the skills to describe the $f_2(x_0, x_1, x_2)$, so it will not be discussed in this aspect. The threshold gates are relatively simple, which means more variables and nodes must be employed to describe Boolean functions. For example, $f_2(x_0, x_1, x_2)$ is described in disjunctive form, in which two different values (x_i, x_i') $(i = 0, 1, 2)$ of each variable are located in multi parts of the given disjunction. Since the two values cannot be represented by a single variable of threshold gates, the number of variables and nodes used to construct the structure must be multiplied. Besides, since the reciprocal relationship between x_i and x_i' is broken, x_i and x_i' are two entirely independent variables, which means the NOT operation cannot be supported by threshold gates.

Since the nodes numbered 0 and 1 are deleted to reduce the storage cost, the number of nodes in OBDD

TABLE 2. Efficiency analysis and comparison of CP-ABE scheme.

Scheme \ Indicator	Encrypt	KeyGen	Decrypt		Ciphertext Size	Secret Key Size
	E_{G_0}	E_{G_0}	E_{G_1}	P_e		
LN [5]	N+1	$2N$+1	O(N)	O(N)	$(N+1)\cdot B_{G_0}+B_{G_1}$	$(2N+1)\cdot B_{G_0}$
BSW [2]	2Φ+1	$2l$+2	O(σ)	O(σ)	$(2\Phi+1)\cdot B_{G_0}+B_{G_1}$	$(2l+1)\cdot B_{G_0}$
RD [9]	2Φ+1	l+3	O(σ)	O(σ)	$(2\Phi+1)\cdot B_{G_0}+B_{G_1}$	$(l+2)\cdot B_{G_0}$
BK[6]	Φ+1	l+1	O(σ)	O(σ)	$(\Phi+1)\cdot B_{G_0}+B_{G_1}$	$(l+1)\cdot B_{G_0}$
Our scheme	T+1	2	O(1)	O(1)	$(T+1)\cdot B_{G_0}+B_{G_1}$	$2B_{G_0}$

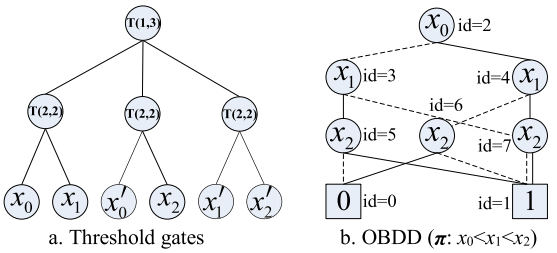


FIGURE 4. Representations of $f_2(x_0, x_1, x_2)$.

representation is less than that of threshold gates representation. More importantly, only three variables are needed, only half of that in threshold gates representation.

To give a more intuitive comparison, the Table 1 is listed below, which indicates that our scheme can not only support all kinds of Boolean operations but also achieves a much better overall performance.

In terms of further measuring the performance in encryption and decryption of CP-ABE, the following indicators are used: complexity of key generation, encryption and decryption, ciphertext size and secret key size. In the measurement of complexity, the number of exponentiation steps in G_0/G_1 and the number of bilinear pairings have more significance, since these operations require much more time than other operations in both encryption and decryption [2].

In Table 2, the meaning of each symbol is as follows: E_{G_0} and E_{G_1} represent the exponentiation number in G_0 and G_1, respectively, $P_e$is the number of bilinear pairings computation,$N$is the number of global attributes, $\Phi$is the number of attributes contained in the access structure, l is the number of attributes used to generate private user keys, $T$is the number of valid paths contained in our OBDD access structure, σ is the least number of attributes used to decrypt successfully, and B_{G_0} and B_{G_1} represent the size of each element contained in group G_0 and group G_1, respectively.

Table 2 indicates that the new CP-ABE scheme performs better in lots of aspects. Both the time complexity of the ***KeyGen*** algorithm and the ***Decrypt*** algorithm are O(1); in particular, the ***KeyGen*** algorithm only needs two exponentiations in G_0, and the ***Decrypt*** algorithm only needs two exponentiations in G_1 and two bilinear pairings computations. Besides, the size of a secret key is a constant, rather than a function of the number of attributes. These features will greatly reduce the burden of authority in generating secret keys, reduce the communication traffic between authority and decryptor, and realize fast decryption. Besides, the complexity of the ***Encrypt*** algorithm and the size of ciphertext relate to the number of valid paths contained in *OBDD*, rather than the number of attributes; these factors will improve the efficiency in encryption and sharing of ciphertext, especially when T is small.

V. CONCLUSIONS AND FUTURE WORK

Ensuring the security of a CP-ABE scheme and improving its efficiency as much as possible has long been a research hotspot in the field of cryptography. This paper proposes a powerful and efficient CP-ABE scheme based on OBDD. Our scheme supports both positive attributes and negative attributes in the description of access polices, the multiple occurrence of an attribute in the same strategy, and complex access polices by making use of any Boolean operation. Our CP-ABE scheme can resist collision attacks and is proven to be CPA secure. In comparison with several CP-ABE schemes, the new scheme designed in this paper not only improves efficiency and capacity in the expression of access policies, but also reduces the main computation of the ***KeyGen*** algorithm, the size of secret key and the main computation of the ***Decrypt*** algorithm to constants, thus cutting off their relationships with the number of attributes. Besides, the efficiency of the ***Encrypt*** algorithm and the size of ciphertext can also be improved.

In future, we will do more work to enhance the efficiency of the CP-ABE scheme. The OBDD-based access structure and CP-ABE scheme proposed in this paper is a potential work, which can be further studied. For example, by making full use of the technology of the OBDD and applying it to other mechanisms related to the CP-ABE scheme, lots of follow-up work can be realized, such as attribute management, access policy updating, user revocation, and ciphertext updating.

REFERENCES

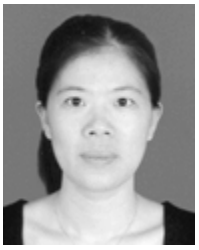
[1] A. Sahai and B. Waters, "Fuzzy identity-based encryption," in *Proc. EUROCRYPT*, Aarhus, Denmark, 2005, pp. 457–473.
[2] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in *Proc. IEEE SP*, Oakland, CA, USA, May 2007, pp. 321–334.
[3] V. Goyal *et al.*, "Attribute-based encryption for fine-grained access control of encrypted data," in *Proc. ACM CCS*, New York, NY, USA, 2006, pp. 89–98.

[4] B. Waters, "Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization," in *Public Key Cryptography—PKC*, Berlin, Germany, 2011, pp. 53–70.
[5] C. Ling and C. Newport, "Provably secure ciphertext policy ABE," in *Proc. ACM CCS*, New York, NY, USA, 2007, pp. 456–465, 2007.
[6] A. Balu and K. Kuppusamy, "An expressive and provably secure ciphertext-policy attribute-based encryption," *Inf. Sci.*, vol. 326, no. 4, pp. 354–362, Aug. 2014.
[7] S. B. Akers, "Binary decision diagrams," *IEEE Trans. Comput.*, vol. 27, no. 6, pp. 509–516, Jun. 1978.
[8] R. Drechsler and D. Sieling, "Binary decision diagrams in theory and practice," *Int. J. Softw. Tools Technol. Trans.*, vol. 3, no. 2, pp. 112–136, May 2001.
[9] Y. S. Rao and R. Dutta, "Dynamic ciphertext-policy attribute-based encryption for expressive access policy," in *Proc. ICDCIT*, Bhubaneswar, India, 2014, pp. 275–286.
[10] 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, Oct. 2013.
[11] H. Deng *et al.*, "Ciphertext-policy hierarchical attribute-based encryption with short ciphertexts," *Inf. Sci.*, vol. 275, pp. 370–384, Aug. 2014.
[12] J. Li *et al.*, "Flexible and fine-grained attribute-based data storage in cloud computing," *IEEE Trans. Services Comput.*, vol. PP, no. 99, p. 1, Jan. 2016.
[13] S. Yu, C. Wang, K. Ren, and W. Lou, "Achieving secure, scalable, and fine-grained data access control in cloud computing," in *Proc. IEEE INFOCOM*, San Diego, CA, USA, Sep. 2010, pp. 1–9.
[14] M. Chase and S. S. M. Chow, "Improving privacy and security in multi-authority attribute-based encryption," in *Proc. ACM CCS*, New York, NY, USA, 2009, pp. 121–130.
[15] J. Li *et al.*, "Multi-authority ciphertext-policy attribute-based encryption with accountability," in *Proc. ASIACCS*, Hong Kong, 2011, pp. 386–390.
[16] M. Li, S. Yu, Y. Zheng, K. Ren, and W. Lou, "Scalable and secure sharing of personal health records in cloud computing using attribute-based encryption," *IEEE Trans. Parallel Distrib. Syst.*, vol. 24, no. 1, pp. 131–143, Jan. 2013.
[17] Z. Wang, D. Huang, Y. Zhu, B. Li, and C.-J. Chung, "Efficient attribute-based comparable data access control," *IEEE Trans. Comput.*, vol. 64, no. 12, pp. 3430–3443, Dec. 2015.
[18] Y. S. Rao, "A secure and efficient ciphertext-policy attribute-based signcryption for personal health records sharing in cloud computing," *Future Generat. Comput. Syst.*, vol. 67, pp. 133–151, Feb. 2017.
[19] W. W. Smari, P. Clemente, and J. F. Lalande, "An extended attribute based access control model with trust and privacy: Application to a collaborative crisis management system," *Future Generat. Comput. Syst.*, vol. 31, no. 1, pp. 147–168, May 2014.
[20] 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.
[21] F. Towhidi, A. H. Lashkari, and R. S. Hosseini, "Binary decision diagram (BDD)," in *Proc. ICFCC*, Kuala Lumpur, MY, USA, 2009, pp. 496–499.

LONG LI received the B.S. degree from North China Electric Power University, Beijing, China, in 2011, the M.S. degree from the Guilin University of Electronic Technology, Guilin, China, in 2014, where he is currently pursuing the Ph.D. degree. His research interests focus on information security, especially the design and analysis of cryptographic algorithms, and access control policies.

TIANLONG GU received the M.Eng. degree from Xidian University, China, in 1987, and the Ph.D. degree from Zhejiang University, China, in 1996. From 1998 to 2002, he was a Research Fellow with the School of Electrical and Computer Engineering, Curtin University of Technology, Australia, and a Post-Doctoral Fellow with the School of Engineering, Murdoch University, Australia. He is currently a Professor with the School of Computer Science and Information Security, Guilin University of Electronic Technology, Guilin, China. His research interests include formal methods, data and knowledge engineering, software engineering, and information security protocol.

LIANG CHANG received the Ph.D. degree in computer science from the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China. He is currently a Professor with the School of Computer Science and Information Security, Guilin University of Electronic Technology, Guilin, China. His research interests include information security, knowledge representation and reasoning, description logics, and the semantic Web.



ZHOUBO XU received the Ph.D. degree from Xidian University, XiŠan China. She is currently an Associate Professor with the School of Computer Science and Information Security, Guilin University of Electronic Technology, Guilin, China. Her research interests include symbolic algorithms, reversible logic and data security in cyber-physical systems, and cloud computing.

YINING LIU received the B.S. degree in applied mathematics from Information Engineering University, Zhengzhou, China, in 1995, the M.S. degree in computer software and theory from the Huazhong University of Science and Technology, Wuhan, China, in 2003, and the Ph.D. degree in mathematics from Hubei University, Wuhan, in 2007. He is currently a Professor with the School of Computer Science and Information Security, Guilin University of Electronic Technology, Guilin, China. His research interests focus on the design and analysis of security protocols.

JUNYAN QIAN received the B.S. degree from the Anhui Polytechnic University, China, in 1996, the M.S. degree from the Guilin University of Electronic Technology, China, in 2000, and the Ph.D. degree from the Southeast University of China, in 2008. He is currently a Professor with the School of Computer Science and Engineering, Guilin University of Electronic Technology. His research interests include formal verification, optimization algorithm, and reconfigurable VLSI design.