CP-ABE based Access Control with Policy Updating and Fast Decryption for Intelligent Manufacturing

Long Li[1], Tianlong Gu[2], Liang Chang[2], Jingjing Li[3], Junyan Qian[2]

[1] School of Electromechanical Engineering, Guilin University of Electronic Technology, China
[2] Guangxi Key Laboratory of Trusted Software, Guilin University of Electronic Technology, China
[3] School of Information and Communication, Guilin University of Electronic Technology, China
lilong@ncepu.edu.cn, {cctlgu, changl}@guet.edu.cn, 1402101004@mails.guet.edu.cn, qjy2000@gmail.com

Abstract

In view of the information security problems existed in intelligent manufacturing system, a system model and its corresponding CP-ABE based access control framework are presented. Moreover, a MA-CP-ABE based on Ordered Binary Decision Diagram (OBDD) is proposed to solve the common drawbacks of CP-ABE schemes, such as limited expression ability of access structure, inefficient sub algorithms, poor system reliability caused by single authority, and lack of policy updating mechanism. The proposed scheme has several remarkable characteristic. Firstly, an OBDD-based access structure is proposed to realize the efficient representation of access policies. Secondly, the sub-algorithms such as key generation algorithm and decryption algorithm have good performance. In particular, the complexity of decryption algorithm is constant, not limited to any variable in the system, which result in fast decryption. Thirdly, the reliability and stability are enhanced by the set-up of multiple attribute authorities. Fourthly, the access policy updating mechanism has better feasibility, because the adoption of proxy ciphertext updating and lazy key generation. Theoretical analysis further proofs the above superiorities of the new scheme. In addition, the new scheme is able to resist collusion attack and provide forward security and CPA security.

Keywords: Intelligent manufacturing, CP-ABE based access control, Binary decision diagram, Access policy updating, Fast decryption

1 Introduction

In recent years, many advanced manufacturing modes or concepts, such as cloud manufacturing [1], industry 4.0 [2] and intelligent manufacturing [3] emerges along with the rapid development of information technologies, e.g., Internet of Things (IoT), cloud computing and artificial intelligence. Especially, the proposal of intelligent manufacturing attracts extensive attentions and triggers discussions about the new industrial revolution [4].

By collecting, processing and transmitting various types of data, intelligent manufacturing can connect some isolated traditional industrial networks, e.g., process control network, industrial IoT and enterprise cloud, into an open industrial internet. During production processing, the open industrial internet can realize intelligent control, information exchange, collaborative sharing, etc. Obviously, cloud platforms [5] are needed to realize data sharing and collaboration. For instance, data files stored in clouds can be retrieved conveniently by other entities through networks. However, many information security problems arises along with the implementation of such kind of systems, including malicious attacks, data theft, unauthorized operation of equipment, collision attack [6-8], etc.

Access control [7] is one of the basic means to ensure the information security of system. It can prevent unauthorized access on purpose. However, many unique characteristics spring up along with the emergence of new networks, such as the one-to-many relationships between entities in intelligent manufacturing network systems [1, 3], the storage and access of personal health records in medical network systems [9] and the anonymous interaction in social networks [10]. Because of these characteristics, the traditional access control strategy become ineffective to complex and changeable networks. Therefore, J. Bethencourt et al. [11] proposed CP-ABE- based access control, which can conduct data encryption and access control simultaneously.

The core idea of CP-ABE is that the data owner formulates the access policy and completes the data encryption, and the data users satisfying the access policy can decrypt the encrypted data. For example, if a data provider sets the access policy as "(*enterprise_1* AND (*project_manager* OR (*general_employee* AND *work_for _ more_than_3_ years*)) OR (*industry_partner* AND *department_ mana- ger*)" and completes the data encryption, the confidentiality of data and the access

*Corresponding Author: Liang Chang; E-mail: changl@guet.edu.cn

DOI: 10.3966/160792642018051903018

control among cooperative enterprises can be achieved at the same time. It can be seen that the CP-ABE is not implemented for a specific data consumer and does not need to obtain the identity information of all data consumers in advance. With the help of common attributes, the CP-ABE can accomplish data encryption and access control, which has good applicability and scalability.

Once CP-ABE is presented, it receives widespread attention and abundant achievements of theoretical and/or practical value have been presented. But along with the continuous development of information technology and user requirements, there are still some problems in CP-ABE deserving further exploration. Firstly, since different network scenarios possess different characteristics, in order to enhance the practicability of CP-ABE, it is necessary to conduct more research under more varieties of network scenarios [12-15]. Secondly, as one the core foundational components of CP-ABE, access structure need to be improved in its expression ability and efficiency [7]. Thirdly, once the single authority in CP-ABE is compromised or blocked, it will inevitably lead to information leakage, system performance degradation or even paralysis. To ensure the safely and smoothly running of the system, multi-authority CP-ABE (MA-CP-ABE) [10, 16-20] should be studied intensively. Fourthly, policy updating mechanism is required to enhance the flexibility and scalability. Fifthly, the complexity of most algorithms contained in CP-ABE are linear to the number of attributes [14, 24], which will decrease system efficiency and bring heavy workloads to users when large number of attributes are contained.

1.1 Motivations and Contributions

According to the analyses above, an MA-CP-ABE based access control that can be applied to intelligent manufacturing is presented in this paper. The design mainly includes two parts: (1) an intelligent manufacturing system model and its access control framework, (2) an efficient MA-CP-ABE scheme that supports policy updating and fast decryption.

The main advantages of this design are: (1) the rational task assignment and cooperation between authorities can enhance the reliability, stability and efficiency of the system; (2) the access structure simplified from Ordered Binary Decision Diagram (OBDD) performs better both in expression capacity and efficiency; (3) the policy updating mechanism can improve the flexibility and scalability of the scheme; especially, the proxy ciphertext updating and lazy key generation can balance the burden of entities; (4) the complexity of the decryption algorithm is constant, which can achieve fast decryption; (5) the complexity of encryption algorithm and ciphertext length are both linear to the number of valid paths in OBDD-based access structure only, and have nothing to do with the number of attributes; (6) the complexity of the key generation algorithm and key size are both linear to the number of authorities, and have nothing to do with the number of attributes; (7) the new scheme can resist collusion attacks, and has forward security and CPA security.

1.2 Organization

The rest of the paper is organized as follows. In Section 2, we introduce some up-to-date literatures about CP-ABE. In Section 3, we review some preliminaries, which will be used in scheme construction and security proof. The system model and its corresponding access control framework are presented in Section 4. The new OBDD- based MA-CP-ABE with policy updating and fast decryption is proposed in Section 5; and the security proof and efficiency analysis are given in Section 6. Finally, we conclude the paper in Section 7.

2 Related Work

At present, the research on CP-ABE has been refining, which concentrates on the performance of access structure, the reliability of whole system, the feasibility of policy updating mechanism, the efficiency of sub algorithms, the application of CP-ABE in different areas, etc.

Access structures in different forms, such as Threshold gates [11], AND-OR gates [17], AND gates [21] and LSSS matrix [22], has been used to design CP-ABE schemes. But these access structures have limited performance, e.g., the positive and negative value of a single attribute cannot be expressed by the same variable and the NOT operation is not supported. OBDD [7] is an ideal choice for CP-ABE access structure. With the help of atomic Boolean operations (namely, AND, OR and NOT), it can not only complete the efficient expression of Boolean function, but has the ability to carry out symbolic operations directly.

The CP-ABE schemes proposed in [7, 11], etc., only containe a single authority, which is responsible for system management and key generation. However, the single authority is not only a performance bottleneck of the CP-ABE scheme, but also an unstable factor in the system. Once it is compromised or blocked, it will inevitably lead to information leakage, system performance degradation or even paralysis. To ensure the safely and efficiently running of the system, MA-CP-ABE [10, 16] are proposed, in which several authorities of diverse assignments cooperate with each other to complete the system management, attribute management, key generation and so on together.

As complex networks emerges and integrates with each other, users and its attributes changes frequently. CP-ABE schemes are required to be flexible enough and have good scalability. Therefore, a CP-ABE scheme with policy updating mechanism is designed in [16].

In CP-ABE schemes, the complexity of algorithms are mostly positively correlated to the number of attributes. Therefore, the more complicated access policy (with more attributes) it is applied, the lower efficiency it will be. This will inevitably limit the application of such schemes in large scale system. By using the concept of decryption outsourcing, Li Q et al. [16] proposed a CP-ABE scheme which can significantly eliminate the decryption overhead for data users. By generating secret keys of constant-size, the CP-ABE proposed in [7] can realize fast decryption.

At present, approaches in literatures mostly assume CP-ABE is applied in cloud computing networks [17, 25], while the studies focus on other network scenarios are relatively less. To provide scalable and efficient video distribution in Information-Centric Networks, ABE is incorporated into Digital Rights Management schemes in [18]; In [10], to guarantee consumers' personal privacy, CP-ABE is embedded in the friend discovery process of mobile social networks; In [12], a unique Secure Health Sensor node which can guarantee the security of sensitive medical data is designed, in which CP-ABE is employed to play the role of providing effective access control. In [9], to assist the secure storage and share of Electronic Medical Records in the cloud, a novel architecture formed by group based CP-ABE and CP-ABE is proposed. In [14-15], ABE is selected to assist the implementation of fog computing and fog communications.

Some literatures devoted to other research points are presented, e.g., the CP-ABE proposed in [19] is adaptively traceable against key-like decryption; a CP-ABE with no trustworthy central authority is proposed in [20].

3 Preliminaries

In this section, preliminaries related to CP-ABE and OBDD are introduced.

3.1 Access Structure

Essentially, an access policy is a rule R, which will return 1 or 0 deterministically according to an attribute set S. In more detail, 1 is returned if and only if S satisfies R, written as $S \vDash R$, and 0 R is returned when S does not satisfy R, written as $S \nvDash R$.

To express access policies intuitively, the concept of access structure is further proposed and several kinds of access structures have been presented, such as AND gates and threshold gates. Each kind of access structure has its unique mathematical forms; for instance, threshold gates describe an access policy as element matching between two attribute sets.

3.2 Bilinear Maps and Bilinear Groups

Definition 1. (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 two properties: (1) Bilinearity: for all $u, v \in G_0$ and $a, b \in_R Z_p$, we have $e(u^a, v^b)=e(u, v)^{ab}$. (2) Non- degeneracy: $e(g, g) \neq 1$.

Definition 2. (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.

3.3 Decisional Bilinear Diffie-Hellman Assumption

Let g be a generator of G_0 and $a, b, z \in_R Z_p$. Let e be a bilinear map e: $G_0 \times G_0 \rightarrow G_1$.The Decisional Bilinear Diffie- Hellman (DBDH) assumption can be described as no probabilistic polynomial-time adversary can tell the difference between $(g, g^a, g^b, g^c, e(g, g)^{abc})$ and $(g, g^a, g^b, g^c, e(g, g)^z)$.

For any probabilistic polynomial-time adversary $\mathbb{A}$, the advantage in solving DBDH problem is defined as $Adv_{\text{DDH}}^{G_0}(\mathbb{A})=|Pr[\mathbb{A}(g, g^a, g^b, g^c, e(g, g)^{abc})=1]-Pr[\mathbb{A}(g, g^a, g^b, g^c, e(g, g)^z)=1]|$. If $Adv_{\text{DDH}}^{G_0}(\mathbb{A})$ is negligible, the DBDH assumption holds.

3.4 Ordered Binary Decision Diagram

Ordered binary decision diagram (OBDD) is a kind of data structure that can be used to achieve representation and manipulation of Boolean functions.

Definition 3. (OBDD) As an expression of the Boolean function $f(x_0, x_1, \ldots, x_{n-1})$, an OBDD 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 OBDD is either non-terminal or terminal.

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

(3) Each terminal node i is labeled with 0 or 1, and has no child node.

(4) Each non-terminal node u has two edges point to *low* and *high* separately; The edge points 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.

(6) On any directed path from the root node to a terminal node, all variables encounter the same order $\boldsymbol{\pi}$.

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

4 Modeling and Access Control of Intelligent Manufacturing System

According to the characteristics of intelligent manufacturing, this paper presents a model of intelligent manufacturing system and specifically designs an access control based on MA-CP-ABE.

4.1 System Modeling

As shown Figure 1, the newly proposed intelligent manufacturing system model mainly contains five types of entities: (1) Intelligent manufacturing cloud service platform (IMCSP) is responsible for global management, such as system management, data storage, establishment of collaborative cooperation, supervision of service process, etc.. (2) Certificate authority (CA) is the core of system information security, and mainly responsible for the generation of public keys and private keys. (3) Attribute authorities (AAs) are established to assist the management of attributes and the generation of secret keys. (4) Service providers (SPs) are responsible for providing kinds of services such as data collecting, products manufacturing. (5) Service consumers (SCs) access to data or equipment according to their access permissions.

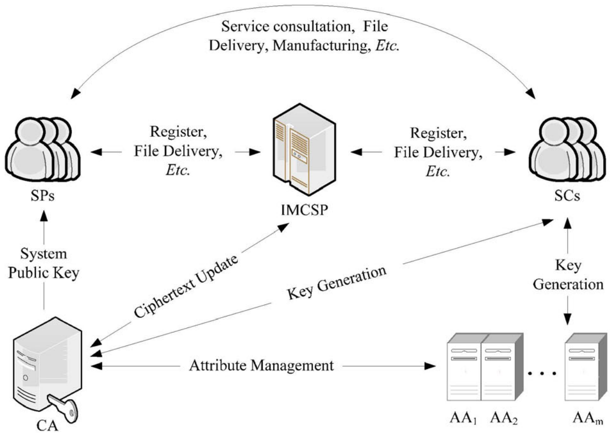

Service consultation, File Delivery, Manufacturing, *Etc.*
SPs
Register, File Delivery, *Etc.*
IMCSP
Register, File Delivery, *Etc.*
SCs
System Public Key
Ciphertext Update
Key Generation
Key Generation
Attribute Management
CA
AA_1
AA_2
AA_m

Figure 1. Intelligent manufacturing system model

4.2 OBDD-based MA-CP-ABE Framework

The OBDD-based MA-CP-ABE presented in this paper consists of two parts: (1) the basic framework of MA-CP-ABE; (2) the policy updating mechanism including proxy ciphertext updating and lazy key generation.

Basic framework. The basic framework of OBDD-based MA-CP-ABE contains following algorithms:

Setup is executed by CA, mainly performs the generation of public key *PK* and master key *MK*.

Encrypt is executed by SPs, mainly performs the encryption of plaintext *M* according a pre-generated OBDD-based access structure.

CAKeygen is executed by CA, mainly performs the generation of a part of DC's secret key *SK*, which is used to bundle all attribute authorities together.

AAKeygen is executed by each AA, mainly performs the generation of a part of *SK* according to *S*.

Decrypt is executed by DCs to decrypt a ciphertext *CT* with their *SK*s.

Policy updating mechanism. The policy updating mechanism consists of two parts: proxy ciphertext updating and lazy key generation.

(1) Four algorithms used to implement proxy ciphertext updating.

SetUpdate is executed to update global attribute set.

ReSetup is executed to run the above ***Setup*** algorithm.

ASUpdate is executed to update the OBDD-based access structure.

CTUpdate is executed by CA to update ciphertext.

(2) Two algorithms used to update the users' private keys.

Determine is executed by DCs to determine whether it is necessary to update their *SK*s;

SKUpdate is executed by DCs, CA and AAs together to re-generate *SK*s.

4.3 Access Control Workflow

The access control in intelligent manufacturing is closely correlated with the system model, which will be illustrated based on the OBDD-based MA-CP-ABE.

Obtainment of access permissions. To ensure the information security, only DCs who have obtained the corresponding permissions can access the data or equipment. This process can be further divided into five steps:

*Step*1: CA executes the ***Setup*** to generate *PK* and *MK*;

*Step*2: SPs set access policies, generate OBDD-based access structures, encrypt files such as data sets, device drivers and control instructions by ***Encrypt***, and upload *CT* to IMCSP;

*Step*3: SCs provide attribute sets to CA and AAs, and receive private keys;

*Step*4: SCs inform IMCSP of their service requirements, and then receive *CT*s provided by IMCSP or SPs;

*Step*5: SCs decrypt *CT*s by ***Decrypt***, and obtain the access permission.

Updating of access permissions. The dynamic changes of users, attributes and access permissions inevitably lead to information security problems. Therefore, access policies, ciphertexts, and *SK*s must be updated in time. This process contains three steps:

*Step*1: SPs set new access policies and generate OBDD-based access structures;

*Step*2: By employing proxy ciphertext updating, CA is assigned to update ciphertexts based on new access structures, old ciphertexts and ***CTUpdate***.

*Step*3: By adopting the idea of lazy key generation, DCs' *SK*s will be updated only when it is needed.

4.4 Safety Indicators and Security Game

Collusion attack resistance. In some cases, multiple authorities or multiple users in a multi-authority ABE scheme may associate with each other trying to decrypt ciphertext that cannot be decrypted individually. Therefore, it is a basic security requirement to resist collusion attacks for any multi-authority ABE scheme. It needs to be noted that CA is considered to be completely credible.

Forward security and backward security. Forward security means the revoked users cannot decrypt newly generated ciphertexts, while backward security means the newcomers cannot decrypt ciphertexts generated before.

CPA security game. Definition 4. (CPA security of MA-CP-ABE scheme) If there is no probabilistic polynomial time within which adversaries can win the following game with non-negligible advantage, the MA-CP-ABE scheme is said to be secure against the chosen plaintext attacks.

The security game between an adversary and a challenger is as follows:

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

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

Phase1, after submitting S, corrupted AAs and a global unique identifier ID, the adversary makes key queries to ***AAKeygen*** and ***CAKeygen*** in MA-CP-ABE, 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 CT. Finally, CT is passed to the adversary.

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

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

In this game, the advantage of adversary A is defined as $Adv^{\mathrm{CPA}}_{\mathrm{CP\text{-}ABE}}$(A)=|$Pr[\mu=\mu']$-1/2]|.

5 OBDD-based MA-CP-ABE

After introducing system model and access control framework, OBDD-based access structure and MA-CP- ABE scheme will be presented.

5.1 OBDD-based Access Structure

By improving the original OBDD, we propose an OBDD-based access structure, aiming at increasing the overall efficiency of the MA-CP-ABE scheme. In this section, theories related to the OBDD-based access structure are presented, such as the definition of valid path and the satisfiability of OBDD-based access structure.

OBDD representation of access policies. Compared with other access structures, such as AND-OR gates, the OBDD has following advantages in the representation of Boolean functions:

(1) Support arbitrary Boolean operations. Since OBDD supports all atomic Boolean operations (AND, OR and NOT), arbitrary Boolean operations can be expressed. For example, the *xor* operation between a and b can be expressed as $(a' \wedge b) \vee (a \wedge b')$.

(2) Both positive and negative value of an attribute can be expressed by a single variable. In an OBDD, each non-terminal node u has two edges. The edge point to left (right) child is called the 0-branch (1-branch), which means the value of the variable contained in u is 0 (1). Therefore, we use $\underline{a}$ to represent a and a' simultaneously.

For any access policy, it can be transformed into a Boolean function, and further represented by an OBDD. Assuming that the access policy has been represented by Boolean function $f(x_0, x_1, \ldots, x_{n\text{-}1})$. According to *Shannon's expansion theorem* $f(x_0, x_1, ..., x_{n\text{-}1})$= x_i· $f_{|\mathrm{x}_i=1}$ + x_i'· $f_{|\mathrm{x}_i=0}$ $(0\leq i\leq n\text{-}1)$, the OBDD representation of $f(x_0, x_1, \ldots, x_{n\text{-}1})$ can be easily obtained by using a recursive process. However, before the OBDD is constructed, the decomposition order of the variables must be defined. This paper assumes that the decomposition order is $\boldsymbol{\pi}$: $x_0<x_1<\ldots<x_{n\text{-}1}$。

Based on *Definition* 3, we further present the following conventions:

The nodes in an OBDD are numbered with integers 0, 1, ..., where 0 and 1 are reserved for terminal nodes $\boxed{0}$ and $\boxed{1}$. The variables contained in non-terminal nodes are numbered with integers 0, 1, ..., n-1. Each non-terminal node can be described by a quadruple <*id*, *i*, *low*, *high* >, in which *id* is node number, *i* is variable number, and *low* (*high*) is used to maintain the relationship to its left (right) child. For the convenience of expression, the above quadruple is simplified as $Node^i_{id}$.

Based on *Shannon's expansion theorem* and above conventions, the algorithm for constructing the OBDD can be descried as follows:

In Algorithm 1, the *computed table* is a dictionary, which is used to store already computed results of previous *Construct* - calls.

After the completion of the above algorithm, we can get $OBDD$={ $Node^i_{id}$ | $id \in ID$, $i \in I$}, where ID is the set of node numbers, I is the set of variable numbers. Since terminal nodes $\boxed{1}$ and $\boxed{0}$ only contain the true or false of Boolean functions, the storage space occupied is negligible.

Algorithm 1. Obtain the OBDD corresponding to a Boolean function

Inputs: A Boolean function f and $\boldsymbol{\pi}$: $x_0<x_1<\ldots<x_{n-1}$
Output: The *OBDD* representation of f

```
node∗ Construct-step(char ∗f, int i);
node∗ Construct(char ∗f) {
  int id=1;
  Empty the computed table;
  return (u = Construct-step(f, 0));
}
node∗ Construct-step(char ∗f, int i) {
  if (∗f == "0")  return($\boxed{0}$);
  if (∗f == "1")  return($\boxed{1}$);
  v0=Construct-step($f|_{x_i=0}$, i+1);
  v1=Construct-step($f|_{x_i=1}$, i+1);
  if computed-table entry (v0, v1,u) exists
 return u;
  u←<++id, i, v0, v1>;
  Store (v0, v1, u) in computed table;
  return(u);
}
```

OBDD-based access structure. A formal OBDD contains all information of a Boolean function. However, when it is used to describe an access policy of ABE, only the information related to encryption and decryption is needed. Therefore, it is necessary to simplify the OBDD to generate a compact and efficient access structure. To describe the rest of the paper more clearly and precisely, we give the below definitions first.

Definition 5. (Valid path) For a directed path from the root node to $\boxed{1}$, if all variables occur according to a predefined ordering $\boldsymbol{\pi}$ and each variable appears no more than one time, the path is called a valid path.

Definition 6. (Satisfiability of an OBDD) After giving S and *OBDD*, the attribute matching is started from the root node following the rules below. For a non-terminal node u and its variable $\underline{a}$, if $(\underline{a}\in S)\wedge(\underline{a}=a)$, the right child is reached along the 1-edge of u. Otherwise, the left child is reached along the 0-edge of u. Repeat the process until one of the terminal nodes is reached. If $\boxed{1}$ is reached, S is considered to satisfy *OBDD*, written as $S \vDash OBDD$. Otherwise, S doesn't satisfy *OBDD*, written as $S \nvDash OBDD$.

Definition 7. (Matchability of a valid path) If $S \vDash OBDD$ and the terminal node $\boxed{1}$ is reached via valid path R, R is considered to be matched with S, written as $S \vDash R$.

The basic idea of converting an OBDD into an OBDD-based access structure is to simplify the structure by removing useless nodes and edges. This process comprises two steps:

*Step*1: Cut off the nodes and edges which are irrelevant to valid paths. That is because only valid paths are useful to the algorithms contained in CP-ABE. This step uses a bottom-up approach started from $\boxed{0}$.

*Step*2: Merge duplicated nodes. *Step*1 is likely to generate some duplicated nodes, which must be merged. This step uses a bottom-up approach started from $\boxed{1}$.

The above steps are both executed from bottom to up, because if a child node of node v can be simplified, node v could be simplified. The pseudocode of the simplification algorithm is shown as Algorithm 2.

Algorithm 2. Simplify an OBDD to an OBDD-base access structure.

Inputs: An OBDD G and its root, $\boldsymbol{\pi}$: $x_0<x_1<\ldots<x_{n-1}$
Output: An OBDD-based access structure

```
void Pruning(node *cn) ;
void Merge(node *cn) ;
node∗ Reduce(node *root) {
  node *Node0←$\boxed{0}$ ;
  Pruning(Node0) ;
  Merge (root) ;
  return root;
}
void Pruning(node *cn) {
  Store parent nodes of cn in P[pn];
  for(i=0; i≤pn; i++) {
    if (P [i].low== cn)    P [i].low←NULL ;
    if (P [i].high== cn)  P [i].high←NULL ;
    if(P [i].low==cn)&&( P [i].high==cn)
        Pruning(node ∗P [i]) ;
  }
}
void Merge (node *v) {
  node ∗final[|G|];
  store all nodes to variable[i] according to i;
  nodeid = 1;
  for ( i=n; i≥0; i--) {
    set an empty list Q;
    for each u in variable[i]   {
      if (u.i == n)  key←(1);
      else  key←(u.low.id, u.high.id);
      add <key, u> to Q;
    }
    Sort the elements in list Q based on their key, the sorting rule is (1)< (NULL,1) < (NULL,2) <…(1, NULL) < (1, 2) <…(2, NULL) < (2, 1) < …;
    existedkey←(0) ;
    while (Q is not empty) do{
      if (key == existedkey)  u.id←nodeid-1;
      else {
        u.id←nodeid;
```

```
            final[nodeid]←u;
            nodeid++;
            existedkey←(u.low.id, u.high.id) ;
        }
      }
    }
}
```

We can see from the above code that the process of constructing OBDD-based access structure mainly contains two functions: (1) Starting from $\boxed{0}$, *Pruning()* gradually cuts off nodes and edges which are irrelevant to valid paths. (2) *Merge()* merges duplicated nodes reassign their *id*s.

Example 1: The generation of an OBDD-based access structure.

An access policy is described in natural language as "*male students live in 10*$^{\#}$ *building, administrators of 10*$^{\#}$ *building, and male administrators of non-10*$^{\#}$ *buildings are authorized to open the anti-theft system of 10*$^{\#}$ *building*".

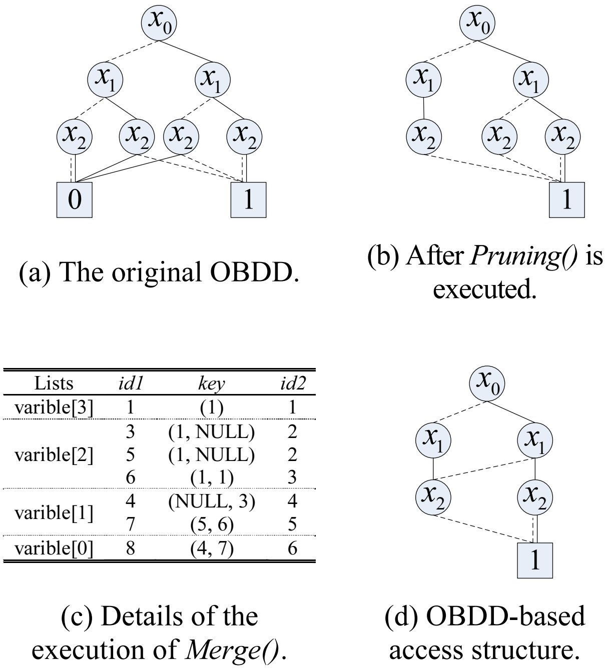
As shown Table 1, "*10*$^{\#}$ *building*", "*male*" and "*student*" are defined as independent attributes, which are represented by x_0, x_1 and x_2 respectively. Therefore, the complementary attributes "*non-10*$^{\#}$ *buildings*", "*female*" and "*administrator*" can be represented by x_0', x_1' and x_2'. The access policy can be expressed as: f= $x_0x_1x_2$+ x_0x_2' + x_0'x_1x_2'.

Table 1. Attributes and variable settings

Attributes	Variables	Attributes	Variables
10$^{\#}$ building	x_0	non-10$^{\#}$ buildings	x_0'
male	x_1	female	x_1'
student	x_2	administrator	x_2'

The original OBDD constructed according to Algorithm 1 is shown in Figure 2(a). The intermediate structure generated by *Pruning()* is shown in Figure 2(b). Some details of nodes during the execution of *Merge()* are shown in Figure 2(c), which includes belonging *list*, original id (*id1*), *key*, and final id (*id2*) of every node. The final OBDD-based access structure is shown in Figure 2(d).

The definitions of valid path, satisfiability of an OBDD and matchability of a valid path, which are listed at the beginning of this section, are applicable to OBDD-based access structure as well. Take the OBDD-based access structure in Figure 2(d) as an example, R$_1$=(x_0→x_1→x_2→ $\boxed{1}$ is a valid path, S$_1$={x_0, x_1} satisfies the structure, S$_1$={x_2} does not satisfy the structure, and S$_1$ ⊨ R$_1$.



(a) The original OBDD.

(b) After *Pruning()* is executed.

Lists	*id1*	*key*	*id2*
varible[3]	1	(1)	1
varible[2]	3	(1, NULL)	2
	5	(1, NULL)	2
	6	(1, 1)	3
varible[1]	4	(NULL, 3)	4
	7	(5, 6)	5
varible[0]	8	(4, 7)	6

(c) Details of the execution of *Merge()*.

(d) OBDD-based access structure.

Figure 2. Simplify OBDD to OBDD-based access structure

5.2 Main Construction of OBDD-based MA-CP-ABE

The newly proposed OBDD-based MA-CP-ABE mainly includes five algorithms:

Setup. CA selects group G_0 of prime order p and group G_1, with bilinear map e: $G_0 \times G_0 \rightarrow G_1$. Then select a generator g of G_0 and a random function $F(s_0, ID) \rightarrow Z_p$, in which $s_0 \in_{\mathrm{R}} Z_p$ and ID is user's identity. CA selects $\{\beta, y, t_1, t_2, ..., t_{|A|}, t_1', t_2', ..., t_{|A|}'\} \in_{\mathrm{R}} Z_p$ and generates public key PK=<g, g^{β}, $e(g, g)^{y\beta}$, $\{(T_j = g^{t_j}$, $T_j' = g^{t_j'}$ $)|1 \leq j \leq |A|\}$> and master key MK=<β, y, $\{(t_j, t_j')|1 \leq j \leq |A|\}$>.

***Encrypt* (*PK*, *M*, *OBDD*)**. Assuming that the OBDD-based access structure includes n valid paths, denoted as $R_{vp}=\{R_1, R_2, \ldots, R_n\}$. DO selects $s \in_{\mathrm{R}} Z_p$, computes $\tilde{C} = M \cdot e(g, g)^{y\beta s}$, $\hat{C} = g^s$ and $C_{R_k} = (\prod_{j \in R_k} \underline{T}_j\ )^s$ $(1 \leq k \leq n)$, in which $\underline{T}_j = T_j$ if $(\underline{j} \in R_k) \wedge (\underline{j} = j)$, else $\underline{T}_j = T_j'$. The ciphertext is CT=<$OBDD, \tilde{C}\ , \hat{C}\ , \{ C_{R_k}\ | R_k \in R_{vp}\}$>.

***CAKeyGen* (*MK*, *ID*$_{\boldsymbol{u}}$)**. Assuming that the system contains m AAs {AA$_1$, AA$_2$, …, AA$_m$}. CA divides A into m non-overlapping parts $\{A_1, A_2, \ldots, A_m\}$, selects $\{\ y_1, y_2, \ldots, y_m\ \} \in_{\mathrm{R}} Z_p$ and sends AA_{SK}^{i} =<A_i, y_i, $\{(t_j, t_j')|j \in A_i\}$> $(1 \leq i \leq m)$ to its corresponding AA$_i$. Then compute $y_u^i = F(y_i, ID_u)$ $(1 \leq i \leq m)$ and generate $D_{CA} = g^{\beta(y - \sum_{i=1}^{m} y_u^i)}$.

***AAKeyGen* ($\boldsymbol{AA_{SK}^{i}}$, *ID*$_{\boldsymbol{u}}$, *S*,)**. AA$_i$ computes $y_u^i = F(y_i,$ $ID_u)$ and $D_u^i = g^{\beta(y_u^i + \sum_{j \in A_i} \underline{t}_j)}$, in which $\underline{t}_j = t_j$ if $(\underline{j} \in S) \wedge (\underline{j} = j)$, else $\underline{t}_j = t_j'$.

Decrypt **(*CT*, { D_u^i |1≤*i*≤*m*}, *D***

Decrypt **(*CT*, { $\boldsymbol{D_u^i}$ |1≤*i*≤*m*}, $\boldsymbol{D_{\mathrm{CA}}}$)**. Suppose DC's attribute set matches R_k. The following calculations are carried out sequentially:

$$D_u = \prod_{i=1}^{m} D_u^i = g^{\beta(\sum_{i=1}^{m} y_u^i + \sum_{j\in A} \underline{t}_j)},$$

$$\hat{D} = D_u \cdot D_{CA} = g^{\beta(y+\sum_{j\in A} \underline{t}_j)},$$

$$D = e(\hat{C},\hat{D}) = e(g,g)^{s\beta(y+\sum_{j\in A} \underline{t}_j)},$$

$$C_k = e(g^{\beta}, C_{R_k}) = e(g,g)^{s\beta(\sum_{j\in R_k} \underline{t}_j)},$$

$$M = \tilde{C} \cdot C_k / D.$$

□

5.3 Access Policy Updating Mechanism

In the practical application of ABE schemes, the users, attributes and access permissions are changing dynamically. In order to cope with this situation and achieve flexible and efficient access control, a practical policy updating mechanism is proposed in this section. This mechanism supports any modification of access policy, such as deletion, addition and replacement of attributes. Besides, by using proxy ciphertext updating and lazy key generation, the workloads of ciphertext updating and key generation are balanced reasonably between entities.

Proxy ciphertext updating.

*Step*1: ***SetUpdate***. If the global attribute set A isn't changed, go to *Step*3; otherwise, update the global attribute set as A^* and go to *Step*2.

*Step*2: ***ReSetup***. CA executes the ***Setup*** algorithm of OBDD-based MA-CP-ABE based on A^*.

*Step*3: ***ASUpdate***. DO updates the OBDD-based access structure as $OBDD^*$ and sends it to CA.

*Step*4: ***CTUpdate*** **(*OBDD**, *CT*)**. According to the n^* valid paths R_k^* ($1\le k\le n^*$) and the attributes contained in each path, CA computes $t_k^* = \sum_{j\in R_k^*} \underline{t}_j$, in which $\underline{t}_j = t_j$ if ($\underline{j} \in R_k^*$)∧($\underline{j}=j$), else $\underline{t}_j = t_j'$. Then calculate $C_{R_k^*} = \hat{C}^{\beta\cdot t_k^*}$ ($1\le k\le n^*$) and update ciphertext as $CT^* = <OBDD^*, \tilde{C}, \hat{C}, \{C_{R_k^*} | 1\le k\le n^*\}>$.

In order to reduce DOs' burden, the above procedure is mostly executed by the powerful authority. Therefore, the updating of ciphertext can also be successfully completed even if DOs are limited in capacities such as computing and storage.

Lazy key generation.

*Step*1: ***Determine***. If the access policy is updated, DC first determines whether the global attribute set is changed. If no change occurs, the *SK* will not be updated. Otherwise, go to *Step*2.

*Step*2: ***SKUpdate***. DC contacts with authorities, and then receives *SK* regenerated by ***AAKeyGen*** and ***CAKeyGen***.

If the lazy key generation is applied, each entity's workload in key updating will be reduced to a certain degree. Because *SK*s need to be updated only when the global attribute set is changed. For other situations, such as the remove of AAs and the change of sub attribute sets governed by AAs, the key re-generation is unnecessary.

Figure 3 summarizes the information interaction between each entity.

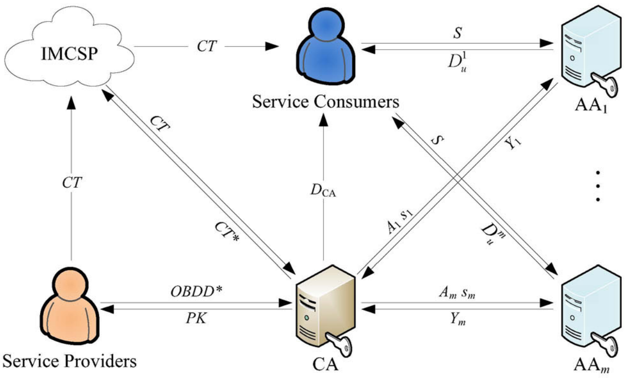


Figure 3. OBDD-based MA-CP-ABE

6 Security and Efficiency Analysis

The correctness analysis is already contained in the decryption algorithm, and is no longer repeated here.

6.1 Security Analysis

Collusion attack resistance.

Claim 1: The OBDD-based MA-CP-ABE proposed in this paper can not only resist collusion attacks launched by multiple DCs, but also resist collusion attacks launched by part of AAs.

Proof:

(1) Collusion attacks launched by multiple DCs.

In a DC's *SK*, an element $y_u^i = F(y_i, ID_u)$ related to ID_u is contained. Since the ID_u is unique to each DC, multiple DCs cannot obtain a valid decryption key by synthesizing their *SK*s. Therefore, the new scheme can resist the collusion attacks launched by multiple DCs.

(2) Collusion attacks launched by part of AAs.

Although part of AAs possess some components of DCs' *SK*s, but they cannot obtain the components generated by CA. Therefore, the collusion attack launched by parts of AAs can be resisted as well. □

Forward security and backward security.

Claim 2: The new proposed MA-CP-ABE scheme is forward secure.

Proof: The access policy updating mechanism can change DCs' authorization status and update ciphertext on demand. Therefore, DCs who have been deauthorized cannot decrypt the newly generated ciphertext. In other words, the scheme is forward secure. □

Meanwhile, we think the backward security is unnecessary for intelligent manufacturing systems. Because whether a *DC* can be authorized or not usually has nothing to do with the time he joins the system. For example, a newly appointed project director must have access to historical files of his project. Furthermore, the system time can be set as one of the attributes if needed.

CPA security.

Theorem 1: If a probabilistic adversary *Adv* can win the CPA security game below with a non-negligible advantage ε, we can construct a simulator *Sim* to solve the DBDH problem with a non-negligible advantage.

Proof:

Selects group G_0 of prime order p and group G_1, with bilinear map e: $G_0 \times G_0 \rightarrow G_1$. *Cha* is the challenger of DBDH problem, he selects $a, b, c, z \in_R Z_p$ and $g \in_R G_0$, and then sends $<g, g^a, g^b, g^c, Z>$ to Simulator *Sim*, in which Z is assigned $e(g, g)^{abc}$ or $e(g, g)^z$ with a same probability. *Sim* will play the role of challenger in the following CPA security game.

Init. *Adv* sends an OBDD-base access structure *OBDD* and an ID_u to *Sim*.

Setup. *Sim* chooses a random function $F(s_0,ID)$, sets $Y=e(g,g)^a$, selects $t_1, t_2, ..., t_{|A|}, t_1', t_2', ..., t_{|A|}' \in_R Z_p$, computes and sends ($T_j = g^{t_j}$, $T_j' = g^{t_j'}$) ($1 \leq j \leq |A|$) to *Adv*.

Phase 1. *Adv* provides S, corrupted attribute authorities and ID_u to makes a secret key query to *Sim*, where $S \nvDash OBDD$. *Sim* divides A into m non-overlapping parts $\{A_1, A_2, \ldots, A_m\}$, chooses numbers β, $y_1, y_2, \ldots, y_m, u_1, u_2, \ldots, u_m \in_R Z_p$, computes $D_{CA} = g^{ab - b\sum_{i=1}^{m} u_i}$.

Assuming AA_1 has not been corrupted, *Sim* computes $D_u^i = g^{b(u_i + \sum_{j \in A_i} \underline{t}_j)}$ ($1 \leq i \leq m$).

Challenge. *Adv* submits M_0 and M_1, which are of equal length. *Sim* selects $u \in_R \{0, 1\}$ and computes $\tilde{C} = M_u \cdot Z$, $\hat{C} = g^c$ and $C_{R_k} = (\prod_{j \in R_k} g^{\underline{t}_j})^c$ ($1 \leq k \leq n$), where $\underline{t}_j = t_j$ if $(\underline{j} \in R_k) \wedge (\underline{j} = j)$, else $\underline{t}_j = t_j'$. The ciphertext is $CT = <OBDD, \tilde{C}, \hat{C}, \{C_{R_k} \mid 1 \leq k \leq n\}>$.

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

Guess. *Adv* guess the value of u is u'. If $u'=u$, *Sim* outputs "DBDH"; otherwise, it outputs "Random".

If $Z=e(g, g)^{ab}$, CT is a valid ciphertext. In this case, *Adv*'s advantage in CPA security game is ε. i.e.

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

If $Z=e(g, g)^z$, CT is absolutely random, which leads the probability of $u'=u$ is 1/2. i.e.

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

In conclusion, *Adv*'s advantage in solving the DBDH problem is $1/2 \cdot (1/2+\varepsilon) + 1/2 \cdot 1/2 - 1/2 = \varepsilon/2$. □

The security comparisons between different schemes are shown in Table 2. It can be seen that most of the schemes are CPA secure, but the analysis about collusion attack resistance, forward security and backward security are missing (represented by "—"). In contrast, the new scheme can resist the collusion attack, is CPA secure and forward secure.

Table 2. Security comparisons

Indicator Scheme	Collision resistance	Forward security	Backward security	CPA security
GW [17]	—	—	—	Yes
MC [24]	Yes	—	—	Yes
XQJ [20]	Yes	—	—	Yes
EQG [10]	—	—	—	—
LTL [7]	Yes	—	—	Yes
Our scheme	Yes	Yes	No	Yes

6.2 Capacity and Efficiency Analysis

By contrasting with similar schemes, this section will analyze the capacity and efficiency of the OBDD-based MA-CP-ABE.

Comparisons of access structures. The comparison of access structures mainly focuses on two aspects: the expression ability and efficiency. As can be seen from Table 3, the newly proposed OBDD-based access structure possesses better performance.

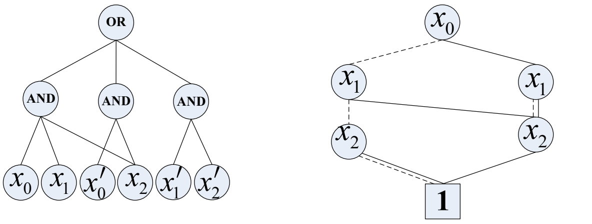
Table 3. Capacities analysis and comparisons between access structures

Indicator Scheme	Access Structure	Operations supported				Number of variables	Number of nodes
		AND	OR	**Threshold**	NOT		
ZDY [23]	AND gates	√	×	×	×	\	\
GW [19]	Threshold gates	√	√	√	×	6	10
Our scheme	OBDD	√	√	√	√	3	5

All the Boolean operations can be implemented by three atomic operations (AND, OR and NOT). Therefore, the expression ability of access structures can be measured by their support for atomic operations. Through simple analysis, it is found that the AND gates only support AND operation; the AND-OR gates and threshold gates support AND and OR operations. These three access structures do not support the NOT operation in essence, because two independent variables are needed to represent the positive and negative value of a single attribute, which cut the inherent relationship between these two values off. By contrast, the OBDD-based access structure proposed in this paper can support AND, OR and NOT operations, which means it is competent in the representation of any Boolean functions.

In terms of expression efficiency, fewer variables and nodes are needed in the OBDD-based access structure. Therefore, the OBDD-based access structure can express access policies more efficiently. Quantitative analysis is further conducted by the following example.

Example 2: Suppose the access policy is represented by $f_2(x_0, x_1, x_2)= x_0x_1x_2+ x_0'x_2 + x_1'x_2'$. As shown in Figure 4 and Table 4, the AND gates cannot represent the access policy, and the AND-OR gates (equivalent to the threshold gates) and the OBDD-based access structure can achieve the representation. Moreover, fewer variables and nodes are required by the OBDD-based access structure.



(a) AND-OR gates

(b) OBDD-based access structure

Figure 4. Access structures of $f_2(x_0, x_1, x_2)$

Comparisons of CP-ABE schemes. Before conducting a detailed comparison of CP-ABE schemes, the frequently used mathematical notations are described in Table 4.

The comparison between OBDD-based MA-CP-ABE and other schemes is shown in Table 5. The new scheme has some advantages in storage space occupation, computation cost and so on.

(1) In the aspect of key generation algorithm, its computation complexity and the size of DC's secret key are not restricted by the number of attributes, but only depend on the number of AAs. Since the number of AAs is far less than the number of attributes, the newly proposed scheme performs better in key generation.

(2) The computation complexity of the decryption algorithm is constant, which is not subjected to any variable in the system. Therefore, the new scheme can realize fast decryption.

Table 4. Notations

Notations	Implications	Notations	Implications
m	Total number of AAs	X	Set of non-leaf nodes
A	Global attribute set	Y	Set of leaf nodes in access structure
n	Number of valid paths	E_{G_0}	Exponentiation number in G_0
S	DC's attribute set	B_{G_0}	Bit length of each element contained in G_0
S_x	Minimum set that satisfies the access structure	P_e	Number of bilinear pairings computation

Table 5. Analysis and Comparisons

Indicator Scheme	*Encrypt*	*KeyGen*	*Decrypt*	Ciphertext	Secret key	Policy updating
	E_{G_0}	E_{G_0}	P_e	B_{G_0}	B_{G_0}	
GW [17]	$2\|A\|+1$	$3\|S\|+m+1$	$2\|S_x\|+2$	$2\|A\|+1$	$2\|S\|+m+1$	Yes
MC [24]	$\|A\|+1$	$\|S\|+1$	$\|X\|+\|Y\|+1$	$\|A\|+1$	$\|S\|+1$	No
XQJ [20]	$\|Y\|+1$	$\|S\|+m$	$\|S\|+2$	$\|Y\|+1$	$\|S\|+m$	Yes
EQG [10]	$2m+\|X\|+2\|Y\|$	$2m+3\|S\|+1$	$\|X\|+2\|Y\|+2m+1$	$2m+\|X\|+2\|Y\|$	$2m+2\|S\|+1$	No
LTL [7]	$n+1$	2	2	$n+1$	2	No
Our scheme	$n+1$	$m+1$	2	$n+1$	$m+1$	Yes

(3) As for the encryption algorithm, its computation complexity and the size of ciphertext have an approximate linear correlation with the number of valid paths in the OBDD-based access structure. It needs to be pointed out that in a practical OBDD-based access structure, the relationship between n and $|A|$ is depicted by $1≤n≤2^{|A|/2}$-1, which is specific to an access policy. As shown in *Example* 2, six attributes is used in AND-OR gates, and five valid paths is contained in OBDD-based access structure.

(4) A policy updating mechanism is designed for the OBDD-based MA-CP-ABE, which can solve any kind of policy updating problem. This mechanism can reduce DOs' burden in ciphertext updating, because the proxy ciphertext updating is applied to distribute most of the computation and storage to cloud server. In addition, each entity's workload in key updating will be reduced to a certain degree, because the lazy key generation is applied, which will update DOs' private keys only when the global attribute set is changed.

7 Conclusions

Advanced manufacturing modes and concepts emerge one after another. But information security issues often exist in such new systems, such as data leakage and illegal remote access of equipment. Therefore, by adopting the idea of joint design, this paper proposes a model of intelligent manufacturing system and its access control framework. The access control is mainly achieved by OBDD-based MA-CP-ABE. Compared with other CP-ABE schemes, the OBDD-based MA-CP-ABE is improved in several aspects, such as performance of access structure, efficiency of sub algorithms and feasibility of policy updating mechanism. In view of its good performance, this approach provides a feasible solution to the information security problem in intelligent manufacturing system.

However, the design is far from perfect, and some problems need to be further studied. For example, the time complexity of encryption algorithm and the ciphertext length are limited by the number of valid paths, the experimental analysis of the system model and access control scheme is lacked. Therefore, in the follow-up work, intelligent manufacturing, OBDD-based access structure and CP-ABE-based access control will be deeply studied in order to increase the security of the system and achieve high performance.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (Nos. U1501252, 61572146, 61562015); the Natural Science Foundation of Guangxi Province (Nos. 2016GXNSFDA380006, 2017GXNSFAA 198283); the Guangxi Key Laboratory of Trusted Software, the High Level of Innovation Team of Colleges and Universities in Guangxi and Outstanding Scholars Program, and the Program for Innovative Research Team of Guilin University of Electronic Technology.

References

[1] L. Ren, L. Zhang, L. Wang, F. Tao, X. Chai, Cloud Manufacturing: Key Characteristics and Applications, *International Journal of Computer Integrated Manufacturing*, Vol. 30, No. 6, pp. 501-515, June, 2017.

[2] C. J. Bartodziej, *The Concept Industry 4.0*, Springer Fachmedien Wiesbaden, 2017.

[3] A. M. Farid, Measures of Reconfigurability and Its Key Characteristics in Intelligent Manufacturing Systems, *Journal of Intelligent Manufacturing*, Vol. 28, No. 2, pp. 353-369, February, 2017.

[4] O. Cardin, F. Ounnar, A. Thomas, D. Trentesaux, Future Industrial Systems: Best Practices of the Intelligent Manufacturing & Services Systems (IMS 2) French Research Group, *IEEE Transactions on Industrial Informatics*, Vol. 13, No. 2, pp. 704-713, April, 2017.

[5] Z. X. Guo, W. K. Wong, C. Guo, A Cloud-based Intelligent Decision-making System for Order Tracking and Allocation in Apparel Manufacturing, *International Journal of Production Research*, Vol. 52, No. 4, pp. 1100-1115, February, 2014.

[6] Y. Liu, Q. Zhong, L. Chang, Z. Xia, D. He, C. Cheng, A Secure Data Backup Scheme Using Multi-Factor Authentication, *IET Information Security*, Vol. 11, No. 5, pp. 250-255, September, 2017.

[7] L. Li, T. Gu, L. Chang, Z. Xu, Y. Liu, J. Qian, A Ciphertext-Policy Attribute-Based Encryption Based on an Ordered Binary Decision Diagram, *IEEE Access*, Vol. 5, pp. 1137-1145, January, 2017.

[8] Y. Liu, G. Liu, C. C. Chang, Lottery Protocol Using Oblivious Transfer Based on ECC, *Journal of Internet Technology*, Vol. 18, No. 2, pp. 279-285, March, 2017.

[9] M. B. Smithamol, S. Rajeswari, Hybrid Solution for Privacy-Preserving Access Control for Healthcare Data, *Advances in Electrical and Computer Engineering*, Vol. 17, No. 2, pp. 31-38, May, 2017.

[10] E. Luo, Q. Liu, G. Wang, Hierarchical Multi-Authority and Attribute-Based Encryption Friend Discovery Scheme in Mobile Social Networks, *IEEE Communications Letters*, Vol. 20, No. 9, pp. 1772-1775, September, 2016.

[11] J. Bethencourt, A. Sahai, B. Waters, Ciphertext- Policy Attribute-Based Encryption, *IEEE Symposium on Security and Privacy*, Berkeley, CA, 2007, pp. 321-334.

[12] D. Sethia, S. Singh, V. Singhal, ABE Based Raspberry Pi Secure Health Sensor (SHS), in: R. El-Azouzi, D. Menasche, E. Sabir, F. De Pellegrini, M. Benjillali (Eds.), *Advances in Ubiquitous Networking 2, Lecture Notes in Electrical Engineering, Vol 397*, Springer, 2017, pp. 599-610.

[13] K.Wang, DACS: DHT-Based Distributed Access- Control System for a Secure Locator/ Identifier Separation Network, *Journal of Internet Technology*, Vol. 17, No. 7, pp. 1413-1421, December, 2016.

[14] Y. Jiang, W. Susilo, Y. Mu, F. Guo, Ciphertext-policy Attribute-based Encryption against Key-delegation Abuse in Fog Computing, *Future Generation Computer Systems*, Vol. 78, No. 2, pp. 720-729, January, 2018.

[15] P. Zhang, Z. Chen, J. K. Liu, K. Liang, H. Liu, An Efficient Access Control Scheme with Outsourcing Capability and Attribute Update for Fog Computing, *Future Generation Computer Systems*, Vol. 78, No. 2, pp. 753-762, January, 2018.

[16] Q. Li, J. Ma, R. Li, X. Liu, J. Xiong, D. Chen, Secure, Efficient and Revocable Multi-authority Access Control System in Cloud Storage, *Computers and Security*, Vol. 59, No. C, pp. 45-59, June, 2016.

[17] G. Wu, Multi-Authority CP-ABE with Policy Update in Cloud Storage, *Journal of Computer Research and Development*, Vol. 53, No. 10, pp. 2393-2399, October, 2016.

[18] J. P. Papanis, S. I. Papapanagiotou, A. S. Mousas, G. V. Lioudakis, D. I. Kaklamani, I. S. Venieris, On the Use of Attribute-Based Encryption for Multimedia Content Protection over Information- Centric Networks, *Transactions on Emerging Telecommunications Technologies*, Vol. 25, No. 4, pp. 422-435, April, 2014.

[19] Z. Liu, Z. Cao, D. S. Wong, Traceable CP-ABE: How to Trace Decryption Devices Found in the Wild, *IEEE Transactions on Information Forensics and Security*, Vol. 10, No. 1, pp. 55-68, January, 2015.

[20] X.-L. Xu, Q.-T. Zhang, J.-L. Zhou, NC-MACPABE: Non-centered Multi-authority Proxy Re-encryption based on CP-ABE for Cloud Storage Systems, *Journal of Central South University*, Vol. 24, No. 4, pp. 807-818, April, 2017.

[21] Z. Wang, D. Huang, Y. Zhu, B. Li, C.-J. Chung, Efficient Attribute-Based Comparable Data Access Control, *IEEE Transactions on Computers*, Vol. 64, No. 12, pp. 3430-3443, December, 2015.

[22] B. Waters, Ciphertext-policy Attribute-based Encryption: An Expressive, Efficient, and Provably Secure Realization. *Public Key Cryptography*, Taormina, Italy, 2011, pp. 53-70.

[23] Z. Wang, W. Liu, CP-ABE with Outsourced Decryption and Directionally Hidden Policy, *Security and Communication Networks*, Vol. 9, No. 14, pp. 2387-2396, September, 2016.

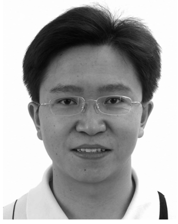
[24] M. Chase, Multi-authority Attribute Based Encryption, *Conference on Theory of Cryptography*, Amsterdam, Netherlands, 2007, pp. 515-534.

[25] S. Tu, Y. Huang, C. M. S. Magurawalage, L. Peng, Z. Zhou, Access Control System Based Cloudlet and ABE on Mobile Cloud, *Journal of Internet Technology*, Vol. 17, No. 7, pp. 1443-1451, December, 2016.

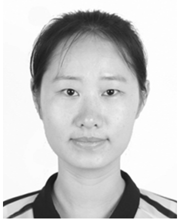
Biographies

Long Li received the BEng degree from North China Electric Power University, Beijing, China, in 2011, the MEng degree from Guilin University of Electronic Technology, Guilin, China, in 2014. He is currently pursuing the Ph.D. degree at Guilin University of Electronic Technology, Guilin 541004, China. His research interests focus on information security, especially the design and analysis of cryptographic algorithms and access control policies.

Tianlong Gu received his Ph.D. degree from Zhejiang University, China. He is 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 his Ph.D. degree from the Chinese Academy of Sciences, Beijing, China. He is a Professor at 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.



Jingjing Li received her BEng degree from Henan Polytechnic University, Jiaozuo, China, in 2010 and obtained her MEng degree from Guilin University of Electronic Technology, Guilin, China, in 2014. Currently, she is a PhD student at Guilin University of Electronic Technology, Guilin, China. Her fields of interest include signal processing and information security.

Junyan Qian received his Ph.D. degree from the Southeast University of China, in 2008. He is a professor at the School of Computer Science and Information Security, Guilin University of Electronic Technology, China. His research interests include formal verification, optimization algorithm, and reconfigurable VLSI design.