JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 33, 837-858 (2017)
DOI: 10.6688/JISE.2017.33.3.15

An Enhanced CP-ABE Based Access Control Algorithm for Point to Multi-Point Communication in Cloud Computing

P. G. SHYNU AND K. JOHN SINGH
School of Information Technology and Engineering
VIT University, Vellore
Tamil Nadu, 632014 India
E-mail: pgshynu@gmail.com; johnsingh.k@vit.ac.in

Ciphertext Policy Attribute-Based Encryption (CP-ABE) is a well-known access control technique, where the data content shared by a single user is accessed by several authorized entities with distinct data access rights, which forms the basis of the point to multi-point communication systems. In such type of systems, the process of user attributes management and user access policy specifications are found to be the challenging tasks. This work introduces an enhanced CP-ABE based access control algorithm, which extracts meaningful user attributes from a set of the user given attributes and stores it over a separate database system. Based upon the extracted attributes, a complete user attribute hierarchical access structure is framed. The user access is provided only when the user attributes hierarchical access structure satisfies the user access policy defined by the data owner. In this manner, the proposed system better solves the drawbacks of existing systems. Experiments show that our proposed algorithm provides comparatively efficient user access than existing CP-ABE techniques and it has lesser computational complexity.

Keywords: ciphertext policy attribute based encryption, access control, user attribute hierarchical access structure, core and reduct attributes, cloud security

1. INTRODUCTION

Cloud computing is a ubiquitous term, provides on-demand services in a flexible manner; it outsources users sensitive data to semi-trusted cloud servers addressing the issue of user's data privacy to greater importance. Since the cloud servers do not only include trusted domain of data users, the technique of encryption and access control are desirable to protect the data content. Attribute-based encryption (ABE) is a widely adopted access control method, used for data protection in cloud computing environment [1-4]. It offers 'one to many' encryption services; that is the point to multi-point communication systems, which enables a single encrypted file to be decrypted among multiple prospective recipients. At present, the existing ABE techniques assume equal privileges to digital contents and authorized users, but the emerging application scenarios demand different rights to both the digital content and cloud data users [5]. Ciphertext Policy Attribute-Based Encryption (CP-ABE) is an emerging technique, where each attribute describes a user entity and each entity can possess n number of attributes [6-8]. Encryptors specify access policy over the ciphertext and share a message with a group of users. A decryptor can get access to a message only when their attributes satisfy the access policy [9, 10]. The key feature of the CP-ABE scheme is that it offers different user access levels depending upon the set of attributes possessed by them, which makes its

Received July 18, 2016; revised October 8, 2016; accepted October 10, 2016.
Communicated by Ram Chakka.

application appropriate to point to multi-point systems. These unique properties enable the use of CP-ABE techniques among several systems, especially with systems where significant data access control is required, for a large number of users [11]. The process of user attribute management and user access policy specification remains to be the two major problems of a point to multi-point systems.

In recent years, we come across databases in which, when the number of objects (users) gets larger and their dimensionality (number of user attributes) come to be larger as well. The number of attributes may increase gradually from tens to hundreds and even to thousands, in many real worlds, point to multi-point, multi privileged cloud applications [12, 13]. Attributes that are irrelevant to user access provision may sometimes deteriorate the performance of cloud system [14]. Likewise storing and processing of all the user attributes including both the relevant and irrelevant attributes is computationally expensive and impractical. In other words, as there exists n number of data users with distinct access privileges, the process of maintenance of all the user attributes is highly complex and challenging. So the process of user attribute management in CP-ABE systems are very much complicated. Further, in CP-ABE the data users acquire their private keys only after the data owner encrypts the data content on data access policies. The data encrypted by the data owner without the knowledge of the actual group of data users, who can decrypt the data but they only specify the data access policies, which allow them to decrypt the data. This may sometimes reduce the level of fine-grained user access provision. Also, when there exists, a significant number of user attributes, the process of user access policy specification is highly complicated as the ciphertext length may increase linearly with an increasing number of user attributes specified in data access policy. This obscures the process of data access policy specification and user access provision in CP-ABE system. K out of n attributes, conjunction and disjunction, are the three top techniques [7-10] through which the existing CP-ABE techniques defines their access policies over the ciphertext.

1.1 Our Contributions

In this paper, we propose a novel CP-ABE based access control algorithm construction, which solves the problem of user attribute management, user access policy specification and fine-grained access provision. The proposed algorithm consists of four major steps, described as follows:

1. *Elimination of Dependent Attributes*: The first phase associated with the proposed algorithm is that it identifies and eliminates the dependent attributes from the system. For example, the attribute age can be derived from the attribute Date of Birth. In that case age is a dependent attribute and it is to be eliminated from the system.
2. *Reduction of User Attributes*: Once the dependent attributes are eliminated, the proposed algorithm computes core and reduct attributes and stores it in a separate database.
3. *Formation of User Attribute Hierarchical Access Structure*: Next a user attribute hierarchical access structure is framed using the core and reduct attributes. The user attribute hierarchical access structure arranges user in a hierarchical manner depending upon the set of attributes possessed by them.

4. *User Access Policy Specification and Verification*: In the proposed model the data owner specifies their access policy using *K* out of *n* attribute technique. The data owner specifies *k* number of attributes and states the user who possesses *n* out of those attributes can access the data. User access is granted, only when the user given attributes matches with the data owner defined access policy. The *K* out of *n* attribute technique used in our proposed system is collision resistant as the user attributes are already extracted in a meaningful way and arranged in a hierarchical order using user attribute hierarchical access structure.

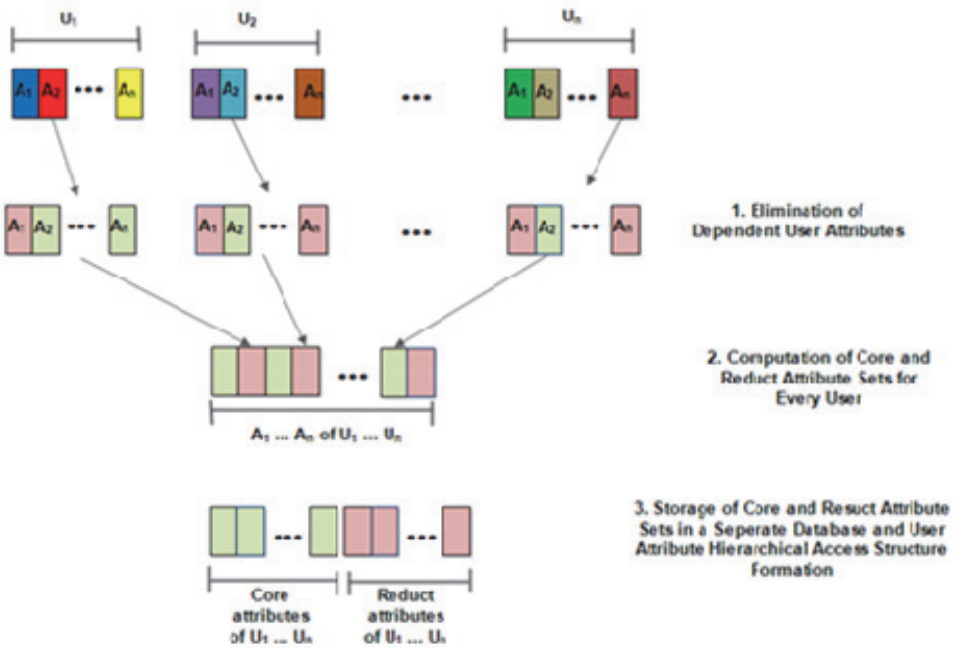


Fig. 1. Working of the proposed system.

As the proposed algorithm extracts and maintains meaningful attributes in a separate database, the process of user attribute management has made easier and effective. Further, it solves the problem of user access policy specification through the identification of core and reduct attributes. This is because of the reason that the data owner defines data access policy using the sensitive user attributes, and it is refined in the proposed algorithm. This gradually reduces the increase in the size of the ciphertext with respect to the number of user attributes defined in data access policy. The working of the proposed scheme is clearly illustrated in Fig. 1.

The remainder of the paper is organised as follows: section 1 introduces the paper; section 2 discusses the works that are related to the proposed scheme and the preliminaries required for the proposed system is described in section 3. A brief explanation of the proposed algorithm and the experimental results are given in section 4. Section 5 concludes the paper.

2. RELATED WORK

The concept of CP-ABE was first given by [15]. In traditional ABE techniques, the process of user access provision is given only when the data user possesses a certain set of user attributes [6]. Such type of access policies depends on the trusted cloud server to store and process user given attributes, to provide user access provision, which leads to data confidentiality issues. This work solves this issue through a CP-ABE technique that realizes complex access control policies and solves data confidentiality issues even at untrusted cloud servers. In this work, the data owner specifies the data access policies and the users of the system are defined using their attributes. The process of user access provision is given when the data owner defined access policies satisfy the user given attributes. This method is almost closer to traditional access control policies.

A bounded CP-ABE scheme was given by [7] in the year 2008 that provides a CP-ABE scheme with advanced access structure. In this work, the user access structure is represented using bounded size access tree with threshold gates as its nodes. The size of the bounded access tree is chosen during the system set up phase. The security proof of this work is based on standard Decisional Bilinear Diffie-Hellman assumptions. The existing CP-ABE techniques support only limited access structures and this work solves this limitation through the use of bounded access trees. However, the size of the ciphertext gradually increases with increase in the number of user attributes, which is a major drawback in this scheme as well as the existing CP-ABE techniques [10].

In order to solve the above issue [9] provides a CP-ABE scheme with constant ciphertext length. This work solves this issue by making the size of the ciphertext be constant with fixed number of pairing computations. Following this work an expressive, efficient CP-ABE scheme was given by [8]. It allows the data owner to specify data access policies using any access formulas over attributes in the system. The ciphertext size, encryption and decryption time scales linearly and depends on the complexity of the access formula. Further, it presents three constructions within a framework and it is provably secure under Parallel Bilinear-Diffie-Hellman Exponent (PBDHE), weaker Bilinear-Diffie-Hellman Exponent and decisional Bilinear-Diffie-Hellman assumptions.

In [16], an extension to traditional proxy re-encryption scheme is given and it enables a semi-trusted proxy server that converts a ciphertext with an access policy to one with the same plain text under another access policy (proxy re-encryption) without providing any information about the underlying plaintext to the proxy server. This system is found to be secure and is widely adopted for several applications. Further in [17], an efficient secure and forward secure CP-ABE scheme is given. This work makes use of a new cryptographic primitive called ciphertext policy attribute-based encryption with augmented hierarchy, through which it solves the problem of user attribute management and access policy specification. It is based on hierarchical identity-based encryption and CP-ABE properties. The security proof is given under three complexity assumptions. It further provides improved system efficiency measures but it fails to provide fine-grained access control properties.

However, the problem of increasing ciphertext length remains to be a major drawback in the existing CP-ABE techniques, which makes its application difficult among communication constrained environments [18]. Another approach of the constant sized CP-ABE scheme is given in [19], which considerably decreases the size of the ciphertext

to be constant with respect to the access policies defined using AND gates with any number of the user given attributes. Each constant sized ciphertext is defined using two elements on a bilinear group. Further, it applies this technique to broadcast encryption systems where the data owner defines their access policies without explicitly specifying the set of data users. The major advantage of this technique is that it reduces the computational overheads to $O(\log N)$. However, the data owner defines their access policies without having any knowledge about the actual set of data users which may lead to reduced fine-grained access control measures. Following this work, in [20], an efficient privacy preserving CP-ABE scheme is given. The primary objective of this work is reducing the increasing size of the ciphertext with respect to the increasing number of attributes. This work reduces the size of the ciphertext to be constant with any number of attributes. Further, it leverages hidden access policy construction through which the privacy of the system users is preserved. In [21] a ciphertext policy attribute-based broadcast encryption scheme is given, where the user attributes are attached to their private keys and an index while ciphertext is accompanied with an access structure and broadcast set. A user can decrypt a ciphertext if and only if the attributes in the user private key satisfies the data access policy and its index is the part of the broadcast set. In this manner, this work provides secure user access provision.

It has been clearly envisioned from the literature [22-26] that the existing CP-ABE schemes provide efficient user access provision with expressive data access policies, however, its application to point to multi-point systems lags at fine-grained access control measure. This is due to the reason that the existing CP-ABE techniques cannot efficiently manage the increasing ciphertext size with respect to the increase in a number of user attributes, which results in the problem of user attribute management and user access policy specification in CP-ABE schemes. Our proposed scheme describes an effective solution to user attribute management and user access policy specification in CP-ABE techniques with fine-grained access control properties.

3. PRELIMINARIES

This section clearly states the preliminaries required for the construction of the proposed system.

3.1 Decision Tables

The information about the cloud data user is represented in the form of an information system, which is also called as decision table. It is formally represented as $IS = (U, A)$, where U is a finite non-empty set of users (instances or rows) and A denotes a finite non-empty set of user attributes. Such that for every $a \in A$, $a: U \rightarrow V_a$, where V_a is the set of values of the attributes A. A decision system $A=(U, A \cup d)$, is a special kind of information system; used in the context of classification; where d is the decision attribute.

3.2 Indiscernibility Relations

Given an information system $IS=(U, A)$, for any $F \subseteq E$ then the equivalence relation R_F is defined as $R_F(x, y)=\{(x, y) \in U^2 | \forall e \in F, e(x)=f(y)\}$. If $(x, y) \in R_F(x, y)$, then X and Y

have exactly the same values for attributes in F. The equivalence relations between these classes are called as indiscernibility relations and it is denoted as $[x]_F$.

3.3 Approximations

Given that $F \subseteq E$, then $X \subseteq U$ is approximated by the use of information in F through the construction of F-lower and F-upper approximation of X. The lower approximation is the union of all equivalence classes and it provides a complete set of users, which belongs to a target set with full certainty. Similarly, the upper approximation represents the possible members of the target set.

$R_f \downarrow X = \{X \in U | [X]_F \subseteq X\}$;
$R_f \uparrow X = \{X \in U | [X]_F \cap X \neq 0$.

3.4 Positive Region

The positive region represents the fuzzy set of objects in U, which can be patently classified using the conditional attributes of F.

$POS_F = \cup_{X \in U} R_F \downarrow [x]_f$

3.5 Boundary Region

The boundary region defines the set of users, where $X \in U$; then it can be probably classified using the conditional attributes of F without any certainty.

$BND_F = \cup_{X \in U} R_F \uparrow [x]_d \setminus \cup_{X \in U} R_F \downarrow [X]_d$

3.6 Reduct

Reduct defines the subset of conditional attribute set that functions similar to the original conditional attribute set without any loss of its classification ability. In simple words, reduct is a subset of conditional attributes.

3.7 Attribute Dependency

The measure to the degree of attribute dependency is represented by k and it measures the level of extent, to which a particular attribute subset is dependent on another attribute subset.

4. PROPOSED SYSTEM

The proposed access control algorithm comprises of two major parts. The first part deals with the problem of the user attribute management process and the second part provides a solution to simplify the process of user access policy specification in CP-ABE schemes.

4.1 User Attribute Management Process in the Proposed CP-ABE System

The proposed algorithm solves the problem of user attribute management through three major steps which are listed as follows:

1. Description to Information system.
2. Elimination of dependent attributes.
3. Identification of reduct and core attributes.

The above-mentioned steps are clearly elaborated in the fourth coming sections and Table 1 provides a sample set of users and the attributes possessed by them. Throughout the paper, this table is taken into consideration for illustration purposes. For the simplicity of the process, a description of various user attributes is given at the end of the paper, in Table 4.

Table 1. Sample information system.

Users	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14	A15	A16	A17	A18	A19	A20	A21	A22	A23	A24	A25	A26	A27	A28	Access Level
U1	x	x	x	x	x	x	x		x		x	x	x	x	x	x	x	x	x	x	x	x			x	x		x	High
U2	x	x	x			x	x	x	x	x		x			x	x		x			x		x		x	x	x	x	Medium
U3	x			x	x			x	x		x	x	x	x			x	x		x		x		x	x		x	x	High
U4	x	x	x		x	x																					x	x	Denied
U5	x		x	x	x	x	x		x			x		x	x	x		x	x	x		x			x	x	x	x	Medium
U6	x	x	x	x			x	x	x	x		x	x	x			x	x		x		x			x	x	x	x	Medium
U7	x		x	x																					x		x	x	Low

4.1.1 Description of information system

Let us consider an information system for the set of users $U = \{u_1, u_2, u_3, ..., u_n\}$ with attributes $A = \{a_1, a_2, a_3, ..., a_n\}$ then the information system IS is defined as IS = $\{U, C_a, D_a, V_a, D_f\}$, where U is the non-empty finite set of users, which forms the universal set. C_a and D_a are the finite set of conditional and decisional attributes such that $(C_a \cup D_a \neq 0)$ and $(C_a \cap D_a = 0)$. For every $a \in A$, $a: U \rightarrow V_a$, where V_a is the value set of a that defines the value of each attribute associated with the system. The decision associated with the information system IS is defined as DS = $(U, C_a \cup \{d_a\})$. A system can have one or more decision attributes in our case a single decision attribute is taken in to consideration. D_f is the decision function that represent the relationship between the users with respect to their attributes D_f: $U \times (C_a \cup D_a) \rightarrow V_a$ such that $D_f(u, a) \in V_a$, for every $a \in C_a \cup D_a$ and $u \in U$.

Such that Table 1 defines an information system that represents a set of system users with respect to their attributes through which the decision of level of user access provision can be made.

4.1.2 Detection and elimination of dependent user attribute

One of the problems with the process of user attribute management is to identify the set of conditional attributes that uniquely determines the set of decisions to be performed if the condition is satisfied. Let $IS = (U, C_a, D_a, V_a, D_f)$ be a decision table, and let G, $H \subseteq C_a \cup D_a$. We then define that H depends in a degree K, $(0 \leq K \leq 1)$ on G in IS if $K\gamma G(H)$. This is symbolically defined as $G \xrightarrow{K} H$. If $K=1$ it defines that H is totally dependent on G or in short depends. If $0 \leq K \leq 1$, then H is partially dependent on G and if $k = 0$ then H is completely independent of G. The relationship $G \xrightarrow{J} H$ can also be written as $G \rightarrow H$. Such that $G \xrightarrow{J} H$, if and only if $A^{\sim} \subseteq C^{\sim}$.

The following are the two important definitions that assist in the determination of dependency between attributes.

Definition 4.1: Consistency of an Information System; An information system; $IS = (U, C_a, D_a, V_a, D_f)$ is consistent that is deterministic if and only if $C_a \rightarrow D_a$.

Definition 4.2: Decomposition of an Information System; $IS = (U, C_a, D_a, V_a, D_f)$ can be uniquely decomposed in to two Information Systems such as $IS_1 = (U_1, C_{a1}, D_{a1}, V_{a1}, D_{f1})$ and $IS_2 = (U_2, C_{a2}, D_{a2}, V_{a2}, D_{f2})$ Such that $C_a \rightarrow D_a$ in IS_1 and $C_a \rightarrow D_a$ in IS_2, where $U_1 = POS_{Ca}(D_a^*), f_1$ is the restriction of f to U_1 and $U_2 = \cup_{u \in Da^*} G_n C_a(u)$.

The expression $POS_{Ca}(D_a)$, defines the positive region of the partition $U|D_a$ with respect to c_a, which means that all the users of the set U can be uniquely classified into blocks of partition $U|D_a$ by means of C_a. D_{f2} is the restriction of the decision function D_f to U_2, and V_{a1}, V_{a2} are the range of function D_{f1} and D_{f2} respectively.

Remark 1: In this manner, the dependent attributes are eliminated from the system. In the case of necessity, the eliminated dependent attributes can be obtained from the users of the system, through user registration phase or setup phase.

Algorithm 1: Algorithm to Compute Dependent Attributes

Input: Decision table $IS = (U, C_a, D_a, V_a, D_f)$
Output: Dependent User Attributes $dep \rightarrow \Phi$
(*dep* is the pool to store dependent attributes);
while$(G, H \subset C_a \cup D_a)$ **do**
Compute K;
$K(0 \leq K \leq 1) \leftarrow$Dep$(H \rightarrow G)$;
if $K=1$ **then**
Total$\leftarrow$Dep$(G \rightarrow H)$;
else if $(0<k<1)$ **then**
partial$\leftarrow$Dep$(G \rightarrow H)$;
else null$\leftarrow$Dep$(G \rightarrow H)$;
end
return $<k=\gamma_G(H)$; where$(G \rightarrow^K H)>$
end

Example 1: In Table 1 there exist n number of user attributes, but there is no assurance that all the attributes are required for the process of user access provision. So, let us detect and eliminate the dependent attributes from the system using the above-defined steps. Let us find out the dependency between attributes of the information system given in Table 1. The attributes $\{a_1, a_2, a_3, \ldots, a_{28}\} \Rightarrow$ {High-level access}, we get K=2/6=1/3, because only 2 out of six users can be uniquely classified as having high-level access or not with the attributes $a_1, a_2, \ldots, a_{28}$.

Similarly K value for mid-level, low-level and denied access with respect to attributes $a_1, a_2, a_3, \ldots, a_{28}$ are,

Mid-level k = 2/6 =1/3;
Low-level k =1/6;
Access denied k = 1/6;

Now let us find how exactly the high level access can be provided by only using the attributes in upper approximation, which includes the attributes $\{A_1, A_2, A_3, A_4, A_7, A_8, A_9, A_{10}, A_{11}, A_{12}, A_{13}, A_{14}, A_{16}, A_{17}, A_{18}, A_{19}, A_{20}, A_{21}, A_{22}, A_{23}, A_{24}\} \rightarrow$ {High level access}, we get K=2/6=1/3. Similarly for,

Midlevel K = 2/6 =1/3;
Low-level K = 1/6;
Access Denied K = 1/6.

Similarly, in all the three cases the attributes $\{A_5, A_6, A_{15}, A_{25}, A_{26}, A_{27}, A_{28}\}$ is no way related to the decision attribute access level. Hence these attributes can be eliminated from the system and it is obtained from the user when needed. Likewise, the iterations are repeated for a different combination of user attributes (which includes attributes in the boundary region, lower and upper approximation regions). Table 2 represent a sample information system defined in Table 1 with no dependent attributes.

Table 2. Information system with no dependent attributes.

Users	A1	A2	A3	A4	A7	A8	A9	A10	A11	A12	A13	A14	A16	A17	A18	A19	A20	A21	A22	A23	A24	Access Level
U1	x	x	x	x	x		x		x	x	x	x	x	x	x	x	x	x	x			High
U2	x	x	x		x	x	x	x		x			x		x			x		x		Medium
U3	x			x		x	x		x	x	x	x		x	x		x		x		x	High
U4	x	x	x																			Denied
U5	x		x	x	x		x			x		x	x		x	x	x		x			Medium
U6	x	x	x	x	x	x	x	x		x	x	x		x	x		x		x			Medium
U7	x		x	x																		Low

4.1.3 Reduction of user attributes

Cloud data systems possess a vast variety of users and their attributes, it has become necessary to know whether all the conditional attributes are necessary to perform the decision of user access provision with respect to data owner specified access policy. Through the process of reduction of attributes the minimal subset of conditional attributes are found without the loss of basic properties provided by the original data set.

Let $IS=(U, C_a, D_a, V_a, D_f)$ be a decision table and let $I \subseteq C_a \cup D_a$ Then the set I is independent in IS, if for every $J \subset I$, $J^{-} \neq I^{-}$; otherwise the set I is dependent. A set $J \subseteq I \subseteq C_a \cup D_a$ is a reduct of I in IS, if J is a maximal independent subset of I.

The above-mentioned reduction property can be briefly explained as follows:

Let $J \subseteq I$ and $i \in J$ then we say that i is dispensable in J if $R(J) = R(J-\{i\})$; otherwise i is indispensable in J. where R denotes the binary relation. If suppose the set J is independent only when all its attributes are indispensable. A subset J' of J is a reduct of J if J' is independent and $R(J') = R(J)$.

Thus reduct is defined as the minimal subset of user attributes, which provides the same set of classification of users as the whole set of attributes.

Algorithm 2: Algorithm to Compute Reduct Set

input: Decision table IS=(U, C_a, D_a, V_a, D_f)
output: One Reduct Set $red \rightarrow \Phi$
(*red* is the pool to store redundant attributes);
while ($J \subseteq I$ and $i \in J$) *do*
Step 1: Compute dispensable relation;
if ($R(J)=R(J-\{i\})$) **then**
i is Dispensable;
else
i is Indispensable;
end
Step 2: Compute dependency of the attribute set;
if ($\forall B$ is Indispensable) **then**
B is independent;
else
B is dependent;
end
Step 3: Compute Reduct;
if (Indep(G' is a reduct of G);
else
G' is not a reduct of G;
end
return <red>
end

The next step in user attribute reduction is to find the core attributes. Let $J \subseteq I$ then the core of J is the set of all indispensable attributes of I. Such that

$$\text{Core}(I) = \cap \text{Red}(J)$$

Here the core is the intersection of all the reduct sets. That is, every attribute of the core belong to some reduct. Thus core is the subset of attributes that contains the most critical attributes for user access provision. This means that any of the attributes present in the core can be eliminated without affecting the classification potential of the user attributes.

To further simplify the information table, some values of attributes are eliminated from the table such that still, we are able to discern objects in the table as like the original one. In this case, the following procedure is applied:

If suppose $i \in J$ then the value of that attribute is dispensable for a user u, if $[u]_{R(J)} = [u]_{R(J-\{i\})}$; otherwise the value of the attribute i is indispensable for the user u. If $\forall i \in J$ the value of the attribute i is indispensable for the user U, then J is considered to be the orthogonal for the user U. If suppose the attribute subset $J' \subseteq J$ is a value reduct of J for the user U, if and only if J' is orthogonal for the user u and $[u]_{I(J)} = [u]_{I(J)}$.

All the indispensable values of the attributes in J for the user u is called the value core of J for the user u, and it is denoted as,

$\text{Core}_u(J) = \cap \text{Red}_u(J)$, where, $\text{Red}_u(J)$ defines the family of all reducts of J for user u.

Algorithm 3: Algorithm to Compute Core and Reduct Sets

Input: Decision table $IS=(U, C_a, D_a, V_a, D_f)$
Output: One Reduct Set and Core Set
red$\rightarrow\Phi$; Core$\rightarrow\Phi$ (red and core are the pool to store redundant and core attributes);
while($i \in G$) ***do***
Step 1: Compute dispensable relation;
if $([u]_R(G))=[u]_{R(G-\{i\})}$) **then**
($i \in G$) is dispensable for u;
else
Indispensable;
end
Step 2: Compute orthogonal for the user u;
if ($\forall(i \in G) \Rightarrow V_i$ is indispensable for u) **then**
orthogonal(u)$\leftarrow J$;
else
J is not an orthogonal of u;
end
Step 3: Compute Reduct;
if (J' is orthogonal for user u and $[u]_R(J)=[u]_{R[J']}$) **then**
$J' \subseteq J \leftarrow$Reduct($J$) for u;
else
$J' \subseteq J$ is not a Reduct for u;
end
return<Red, Which is $\text{Red}^u(J)$>

for ∀Indisp($V_j \in J$) **do**
Compute CORE;
Return<CORE, which is CORE^u(J)>
end

If suppose there exists a dependency $C_a \Rightarrow D_a$, such that the set of decisional attributes D_a do not depend on the complete set of conditional attributes C_a but on its subset C'_a. So it has become necessary to find this subset C'_a. This is done through the use of relative reduct.

Let $D_a \subseteq I$. Then if $C'_a \subseteq C$ is a Decisional-reduct of C_a, the C'_a is the minimal subset of C_a such that, $\gamma(C_a, D_a)=\gamma(C'_a, D_a)$. Then the attribute $i \in C_a$ is Decisional-dispensable in C_a, if POS$C_a(D_a)$=POS($C_a - i(D_a)$; otherwise the attribute i is Decisional-indispensable in C. If suppose all the attributes of i∈C_a then its a conditional-indispensable in C_a, the C_a is called as Decisional-independent. The conditional attributes $C'_a \subseteq C_a$ is a Decisional-reduct of the conditional attribute set C_a if and only if C'_a is decisional independent and POS$C_a(D_a)$=POS$C'_a(D_a)$.

The set of all Decisional-indispensable attributes in C_a is called the Decisional-core and it is denoted by CORE$^{(C_a)}{}_{D_a}$. Then, CORE$D_a(C_a)$=∩Red$D_a(c)$, where Red$D_a(C_a)$ is the family of all decisional reducts of C_a.

Algorithm 4: Algorithm to Compute Decisional-Reducts and Decisional-Core Sets

Input: Decision table IS=(U, C_a, D_a, V_a, D_f)
Output: CORE$D_a(C_a)$, Red $D_a(C_a)$D−*red*→Φ; D−Core→Φ
(*red* and core is the pool to store D−*reduct* and D−CORE *attributes*);
while(C_a, $D_a \subseteq$A) **do**
Step 1: Compute minimal subset of C_a;
if ($C'_a \subseteq C_a$→D−*reduct*(C_a) **then**
C'_a←*minimalsubset*(C_a);
else
C'_a is not a minimal subset of (C_a);
end
Step 2: Compute D−*dispensable* relationship;
if (POS$C_a(D_a)$=POS($C_a - \{i\}(D_a)$) **then**
$i \in C_a$←D−*dispensable*(C_a);
else
$i \in C_a$←D−Indispensable(C_a);
end
Step 3: Compute D−Dependency relation;
if ($\forall i \in C_a$)⇒C_a−*indisp*(C_a)) **then**
C_a←D−*independent*;
else
C_a←D−*dependent*;
end
Step 4:Compute D−Reduct of C_a;
if (C'_a)⇒D−*indep*)&Pos$C_a(D_a)$=POS$C'_a(D_a)$) **then**
$C'_a \subseteq C_a$←D−*reduct*(C);

else
No D−*reduct*(*C*);
end
Return<RedD(*C*)>
Step 5: Compute D−*core* of C_a;
for(∀D−Indispensable($V_a \in C_a$)←D−*core*(C_a)) **do**
Compute D−CORE;
end
Return<(CORE$D_a(C_a)$)>
end

In this manner, the set of core and reducts are calculated for the given set of user attributes. The values of the core attributes are essential attributes and are stored in a separate database and the reducts are also stored separately.

Example 2: In Table 2 we have two relative reducts with respect to high-level access, which are {A_1, A_3, A_7, A_9, A_{11}, A_{13}, A_{12}, A_{16}, A_{18}, A_{20}, A_{22}} and {A_2, A_4, A_8, A_{10}, A_{24}, A_{14}, A_{17}, A_{19}, A_{21}, A_{23}}. Where these two relative reducts are the set of conditional attributes of {A_1, A_3, A_7, A_9, A_{11}, A_{13}, A_{12}, A_{16}, A_{18}, A_{20}, A_{22}, A_2, A_4, A_8, A_{10}, A_{24}, A_{14}, A_{17}, A_{19}, A_{21}, A_{23}}. From this, we can summarize that either the set of the symbolic or numeric attributes are sufficient to make the user access decision. Hence, the set of symbolic attributes are stored in a separate database and it can be extracted when needed. In most of the cases, the symbolic attributes can be derived from numerical values, for example, the attribute branch name can be easily identified from branch id and similarly, the unit name can be derived from unit id by referring the organization database. Hence, the numerical attributes form the core values and the set of symbolic values are identified to be the reduct sets. The core and the reduct attributes are stored separately to reduce the complexity of the system. Core attributes forms the essential set of user attributes without which the process of user access provision is made. Similarly, the reduct and the core set of attributes can be calculated for another decisional attribute mid level access, low level and access denied. In this manner, the set of core and reduct attributes are derived from Table 2 and it is illustrated in Table 3.

Table 3. Set of core and reduct attributes.

User	A_1	A_3	A_7	A_9	A_{11}	A_{12}	A_{13}	A_{16}	A_{18}	A_{20}	A_{22}	Decision
U_1	X	X	X	X	X	X	X	X	X	X	X	High
U_2	X	X	X	X		X		X	X			Mid
U_3	X			X	X	X	X		X	X	X	High
U_4	X	X										Deny
U_5	X	X	X	X		X		X	X	X	X	Mid
U_6	X	X	X	X		X	X		X	X	X	Mid
U_7	X	X										Low

User	A_2	A_4	A_8	A_{10}	A_{24}	A_{14}	A_{17}	A_{19}	A_{21}	A_{23}	Decision
U_1	X	X				X	X	X	X		High
U_2	X		X	X					X	X	Mid
U_3		X	X		X	X	X			X	High
U_4	X										Deny
U_5		X				X		X			Mid
U_6	X	X	X	X		X	X				Mid
U_7		X									Low

Table 4. Attribute description table.

Attribute	Description	Attribute	Description	Attribute	Description
A_1	Employee ID	A_{11}	Performance grade	A_{21}	Host ID
A_2	Employee Name	A_{12}	Overall grade	A_{22}	Chief Name
A_3	Unit ID	A_{13}	Department Name	A_{23}	Chief ID
A_4	Unit Name	A_{14}	Department ID	A_{24}	Service Position
A_5	Date of Birth	A_{15}	Address	A_{25}	Shift time
A_6	Age	A_{16}	City	A_{26}	Onsite employee
A_7	Hired Date	A_{17}	Zip Code	A_{28}	Qualification
A_8	Experience	A_{18}	Manager ID		
A_9	Salary	A_{19}	Manager Name		
A_{10}	Salary grade	A_{20}	Web Address		

Remark 2: By the application of the above-mentioned steps, the proposed algorithm eliminates the dependent attributes and stores the set of core and reducts attributes in a separate database. Hence, only the required attributes are stored in the cloud database, which prevents the user sensitive attributes from various security concerns and assists in effective user attribute storage and retrieval processes. In this way, the proposed algorithm better solves the problem of user attribute management in CP-ABE schemes.

4.2 User Access Policy Specification in the Proposed CP-ABE Scheme

Another problem in the existing CP-ABE technique is the process of user access policy specification, especially with a point to multi-point systems. This is due to the existence of a large number of data users [27, 28]. The proposed algorithm solves the problem of user access policy specification through the formation of user attribute hierarchical structure. A description of user attributes hierarchical structure and the way how it overcomes the problem of user access policy specification in existing CP-ABE techniques are given in the forthcoming sections.

4.2.1 Formation of user attribute hierarchical access structure

Once the set of core and reduct attributes are found, a user attribute hierarchical structure is constructed to reduce the complexity of user attribute management process. It groups the set of related users in terms of the attributes possessed by them. The user attributes hierarchical structure is constructed from two sets U and A, where, U is the set of users and A is the set of attributes possessed by them. Both the core and reduct attribute sets are taken into consideration. The hierarchy arranges the set of users in a relative order depending upon the attributes possessed by them. The upper and lower bounds of the user attributes forms user attribute hierarchical structure.

Let (U, A, I) defines the user attribute hierarchical structure then the two sets U and A has a binary relation between all the elements of the two sets U and A. Thus (A, $\leq$) be a partially ordered set and C be a subset of A. Where C is the set of core attributes. An element r of A with $r \leq c$, for all $c \in C$ is a lower bound of C. Then the upper bound of the C is found dually through the calculation of infimum (inf C or $\wedge$C) or supremum (*sup* C or $\vee$C) of the set C.

The supremum and infimum for an arbitrary set $\{(c_i, r_i) | i \in I\} \subseteq \beta$(U, A, I) is given as

follows:

Supremum: $\wedge_{i\in I}(C_i, r_i)=((\cup_{i\in I}C_i)'', \cap_{i\in I} r_i)$
Infimum: $\vee_{i\in I}(C_i, r_i)=(\cap_{i\in I}C_i, (\cup_{i\in I}B_i)''')$

A complete user attribute hierarchical structure is isomorphic to β(U, A, I) then there exists a mapping $\gamma\tilde{}$:U→L and $\mu\tilde{}$:A→L such that $\gamma\tilde{}$(U) and $\mu\tilde{}$(A) is the supremum-dense and infimum-dense in L, and

$uIa \Leftrightarrow \gamma\tilde{}(U) \leq \mu\tilde{}(A)$

The upper bound of the subset C of partially ordered set (A, ≤) defines all the users with the attributes in the set A which is greater than or equal to all the attributes of C. Similarly the lower bound defines the users with attributes less than or equal to all the attributes of C. Through the use of the upper and lower bound a user attribute hierarchical structure is built.

The user attributes hierarchical structure for sample information system defined in Table 3 is illustrated in Fig. 2.

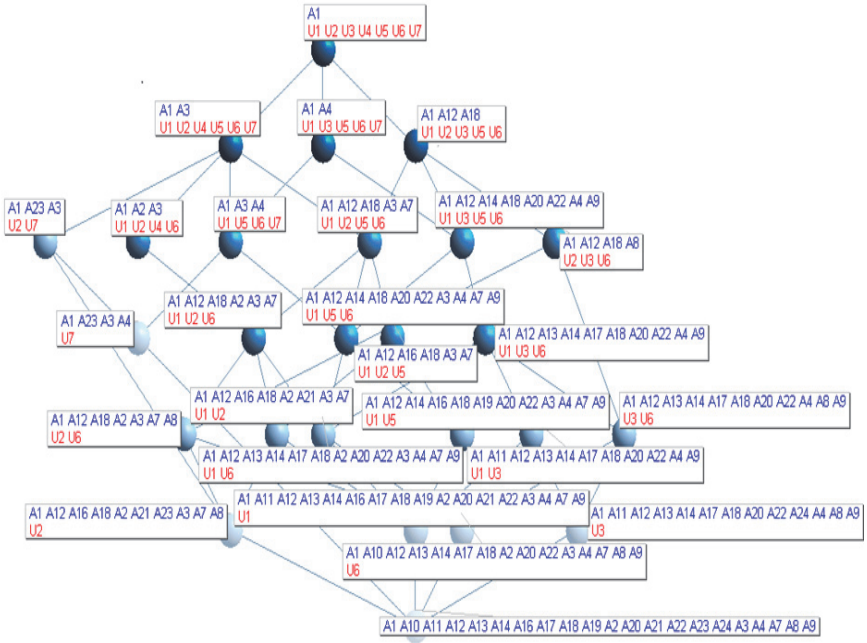


Fig. 2. User attributes hierarchical access structure for a set of users in sample data table.

4.2.2 User access policy specification

Let U={u_1, u_2, …, u_n} be the non-empty finite set of users, which forms the universal set. Then A={a_1, a_2, …, a_n} be the finite set of user attributes, without loss of generality the user with k out of n attributes encrypt a data content to a group of users through

the use of certain user attributes and system's public parameter. The user access is granted only when the data user's credentials satisfy the data owner's access policy. A user access policy is a triplet <F, A, I>, where F and A are the nonempty sets and I is the access relation between F and A, such that I⊆M×A. The elements of F are set of files and the elements of A, the set of attributes. Then if $f \in F$ and $a \in$ A, $<f, a> \in$ I denotes the user access policy, where the file f can be accessed if the user has the attribute a. Similarly, $<f, a> \notin$ I denotes that the file f do not need the attribute a for its access.

Given an information system with n users and m attributes, the data owner defines a corresponding access policy in the form of triplet <F, A, I> consists of a set F = $\{M_1, M_2, ..., M_n\}$ and A=$\{a_1, a_2, ..., a_n\}$ and a access relation I is defined by: $<m_i, A_j> \in$ I. Such that the user can access m_i number of messages if and only if he possesses J out of n attributes. In this manner, the user defines their access policy to a group of users. Whenever the data user tries to decrypt the data content the user-attribute hierarchical structure of a particular user is verified against the data owner defined access policy. User access is granted only when the data owner defined access policy matches with the data user attributes. Since user's attributes are prearranged in a hierarchical manner using user attribute hierarchical access structure the complexity behind user access policy specification and verification processes is reduced.

Example 3: Now let us consider a scenario described in Table 1. Consider an IT organization that grants user access depending upon the set of attributes possessed by them and a member of that organization can be represented by a set of attributes A=$\{a_1, a_2, ..., a_n\}$. Now the CEO of the organization would like to share a file named production norms to all the employees under production department of a particular city with a specific unit name. Such that all the employees under production department of a particular unit and city can access the data file production norms that contain sensitive information about the productivity of the company. In order to access the file production norms the user should be an employee of the organization then they should belong to production department of a particular unit and city. These conditions can be identified through the attributes such as emp-Id or emp-name, unit-Id or unit-name, dep-Id or dep-name and city or zip code. These 4/8 attributes form the basic requirement for user access provision. The data owner specifies that the users with these minimal attributes can access the file production norms and the access is denied for rest of the users. Even though the data file is shared among all the employees with above mentioned 4/8 attributes the data owner likes to provide different levels of data access to the users. Such that the data owner likes to provide high-level access to all the employees who has their manager name as 'XXXX' and chief name as 'YYYY' with overall grade as 'good' and service position as 'senior' with a minimum of '10' years of experience. So in order to provide a high-level access the attributes man-Id or man-name, chief-Id or chief-name, hired date or experience, service position or performance grade and overall grade are required along with the basic attributes for user access provision. Thus 9/17 attributes are sufficient to provide mid-level access. Similarly, the data owner likes to provide a mid-level access to the employees, who have their manager name as 'XXXX' and chief name as 'YYYY' with overall grade as 'good'. The attributes man-Id or man-name, chief-Id or chief-name and over-all grade along with basic user access attributes are required for mid-level access, *i.e.*, 7/13 attributes. A user with basic user access level attributes can perform the Low-level

access.

Now the data owner specifies the access policy as follows:

$U \rightarrow \{A_1, A_2, A_3, A_4, A_{13}, A_{14}, A_{16}, A_{17}, A_{18}, A_{19}, A_{22}, A_{23}, A_7, A_8, A_{12}, A_{24}, A_{11}\} \rightarrow$ {High Level, Production norms} (k=9/17);

$U \rightarrow \{A_1, A_2, A_3, A_4, A_{13}, A_{14}, A_{16}, A_{17}, A_{18}, A_{19}, A_{22}, A_{23}, A_{12}\} \rightarrow$ {Mid Level, Production norms} (k=7/13);

$U \rightarrow \{A_1, A_2, A_3, A_4, A_{13}, A_{14}, A_{16}, A_{17}\} \rightarrow$ {Low level, Production norms} (k=4/8);

In this manner the data owner specifies the user access policy for the set of users and it is illustrated in Fig. 3. In Table 2 the users satisfying K out of n attributes with specified user access policy can access the data.

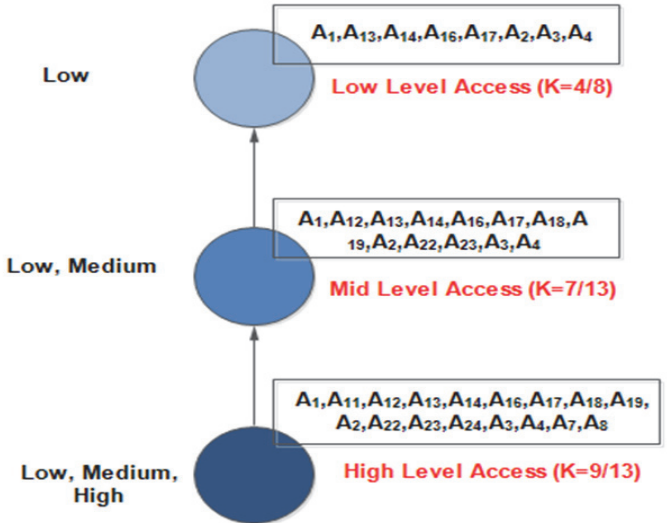


Fig. 3. Data owner defined access policy.

4.3 Results and Discussions

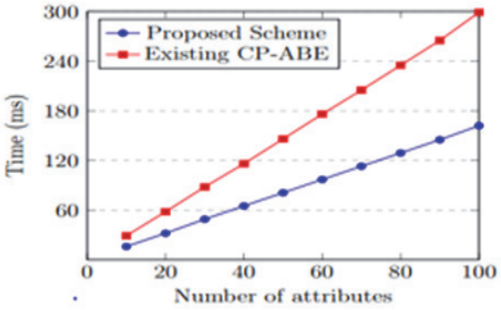
The proposed CP-ABE scheme was implemented using charm crypto, an extensible framework for rapid prototyping systems [29]. To illustrate the characteristics of the proposed CP-ABE scheme as well as to observe its functionality under the real-time environment, we have implemented it over charm crypto, that is installed on an Ubuntu 14.04 LTS 64 bit system at our home institution in the presence of an ambient induced load. The setup is made available for general use to a wider user community as a "public cloud". The configuration of hardware components includes Intel Xenon 3.2GHz processor, 3GB RAM and 40 GB available of single SCSI drive. Charm implementation needs dependencies of c math libraries such as GNU Multi-precision Arithmetic Library, Pairing-Based Cryptography, OpenSSL and Python 3.2. In order to implement the proposed algorithm, the above-mentioned dependencies are installed and executed. Further

224-bit elliptic MNT curve is used in specific. Charm is designed in such a way that it supports the development of advanced cryptographic schemes and protocols. The intensive operations associated with the proposed CP-ABE have been incorporated into c math libraries and it is exposed to charm via python extension using API. The proposed model is applied to self-generated enterprise data set, the status of the sample data is given in the table. The decision table consists of 5670 objects, 28 condition attributes, and one decision attribute, where only a sample of six objects are described in the table for simplicity. First, it finds out the dependent attributes and eliminates it from the system. Next, the core and reduct attributes relating to the system users are computed and stored in a separate database. Through the use of core and reduct attributes a user attribute hierarchical access structure is framed. The process of user access provision is given only when the data owner defined access policy matches with the user attribute hierarchical access structure. In this manner, the proposed system is implemented through charm crypto. The results observed from the experiment have been compared with existing CP-ABE techniques [22].

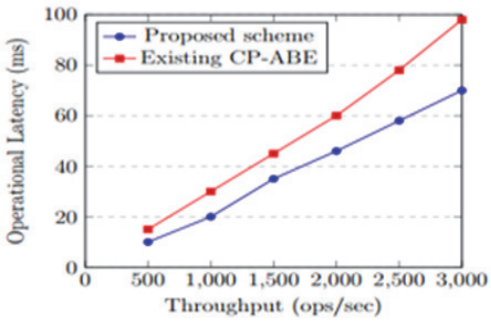
Our experiment focuses on efficiency and performance measures specifically. This is because of the reason that the major objective of this work is to solve the problem of user attribute management and data access policy specification in CP-ABE systems. During the first stage of the experiment, new users have been added to the system using their attributes. Next, in the load phase, nearly 100,000,000 records of 1kb each has been uploaded to the cloud database by a distinct number of data owners and it shared among several groups of data users. Whenever the data owner shares the content among the group of users, a secret key is generated for all the group members. The first experiment which we implemented is designed to measure the secret key generation time. The time taken for secret key generation process is found using the number of user attributes and time in millisecond. It has been observed from the experiment that the proposed algorithm takes an average of 16 milliseconds to compute the secret key for 10 user attributes and nearly 100 attributes have been processed in one second. Whereas, the existing CP-ABE techniques take an average of 29 milliseconds for 10 users attributes. This is because of the reason that the proposed techniques manage user attributes in an efficient manner, through the elimination of dependent attributes and through identification and storage of the core and reduct attributes in separate databases. In this manner, the proposed algorithm better reduces the secret key computation time, in comparison to the existing CP-ABE techniques.

Our second experiment is designed to test the operational latency measure. The average throughput in operations per second and average latency of operations in milliseconds is compared to find the system efficiency measure. Operational latency defines the time to perform an operation such as read, write, update, encrypt and decrypting process. In other words, it defines the time interval between the input and output of the operation. First, a simulation of 95% read operations and 5% write is implemented. That is time taken by the data user perform the read and write operations upon the shared data is calculated. The proposed algorithm demonstrated the best performance with an average latency of 10 milliseconds. It takes a minimum of 10 milliseconds to perform 500 operations and nearly 2600 operations in a second. Whereas, the existing CP-ABE techniques perform only 2000 operations in one minute with an average throughput of 15 to 20 milliseconds. Next simulations consisted of randomly distributed complex read, write and

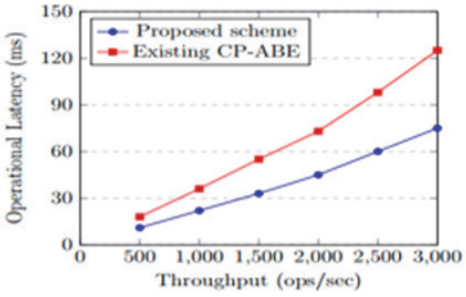
modify operations. The proposed technique showed a maximum throughput of 2500 operations per second. It has an average operational latency of 11 milliseconds for every 500 operations. Whereas the existing technique performs 1800 operations in one second with an average latency of 18 milliseconds for every 500 operations but the average operational latency increases up to 25 milliseconds after 2000 operations. It has been found that the proposed system has better system performance as it simplifies the process of user access policy specification through the implementation of user attribute hierarchical structure.



(a) Secret key generation time.



(b) Operation latency measure-1.



(c) Operational latency measure-2.

Fig. 4. Comparison to efficiency and performance measures.

From the above results it has been clearly envisioned that the proposed CP-ABE scheme has better performance and efficiency measures in comparison to the existing CP-ABE techniques and it is experimentally proven. The lesser the operational latency the lesser is the computational overheads.

5. CONCLUSIONS

In this paper, we investigated how the abilities of the CP-ABE technique can meet the unique requirement of a point to a multi-point communication system in cloud computing environment. Further, the proposed technique better solves the problem of user attribute management and user access policy specification, which are the major barriers

across existing CP-ABE techniques that prevent its application to the larger system with expressive data access control properties. The technique of core and reduct attribute calculation and user attribute hierarchical structure plays a significant role in user attribute management and user access policy specification processes, in the proposed technique. Further, the simulation results show that our novel construction better solves the challenges of existing CP-ABE techniques with inexpensive commodity hardware. In future, this technique can be extended to solve the problem of linearly increasing ciphertext size with increasing number of user attribute.

REFERENCES

1. A. Mishra, R. Mathur, S. Jain, and J. S. Rathore, "Cloud computing security," *International Journal on Recent and Innovation Trends in Computing and Communication*, Vol. 1, 2013, pp. 36-39.
2. D. Jamil and H. Zaki, "Cloud computing security," *International Journal of Engineering Science and Technology*, Vol. 3, 2011, pp. 3478-3483.
3. P. G. Shynu and K. J. Singh, "A comprehensive survey and analysis on access control schemes in cloud environment," *Cybernetics and Information Technologies*, Vol. 16, 2016, pp. 19-38.
4. D. Zissis and D. Lekkas, "Addressing cloud computing security issues," *Future Generation Computer Systems*, Vol. 28, 2012, pp. 583-592.
5. M. D. Ryan, "Cloud computing security: The scientific challenge, and a survey of solutions," *Journal of Systems and Software*, Vol. 86, 2013, pp. 2263-2268.
6. V. Goyal, O. Pandeyy, A. Sahaiz, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in *Proceedings of the 13th ACM Conference on Computer and Communications Security*, 2006, pp. 89-98.
7. V. Goyal, A. Jain, O. Pandey, and A. Sahai, "Bounded ciphertext policy attribute-based encryption," in *Proceedings of the 35th International Colloquium on Automata*, *Languages*, *and Programming*, 2008, pp. 579-591.
8. B. Waters, "Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization," in *Proceedings of International Workshop on Public Key Cryptography*, 2011, pp. 53-70.
9. K. Emura, A. Miyaji, A. Nomura, K. Omote, and M. Soshi, "A ciphertext-policy attribute-based encryption scheme with constant ciphertext length," in *Proceedings of the 5th International Conference on Information Security Practice and Experience*, 2009, pp. 13-23.
10. L. Ibraimi, M. Petkovic, S. Nikova, P. Hartel, and W. Jonker, "Mediated ciphertext-policy attribute-based encryption and its application," in *Information Security Applications*, 2009, pp. 309-323.
11. B. Libert, K. G. Paterson, and E. A. Quaglia, "Anonymous broadcast encryption: Adaptive security and efficient constructions in the standard model," in *Proceedings of International Workshop on Public Key Cryptography*, 2012, pp. 206-224.
12. Z. Zhou, D. Huang, and Z. Wang, "Efficient privacy-preserving ciphertext-policy attribute based-encryption and broadcast encryption," *IEEE Transactions on Computers*, Vol. 64, 2015, pp. 126-138.

13. A. J. Fernández, "Optimum attributes component test plans for *k*-out-of-*n*: F Weibull systems using prior information," *European Journal of Operational Research*, Vol. 240, 2015, pp. 688-696.
14. J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in *Proceedings of IEEE Symposium on Security and Privacy*, 2007, pp. 321-334.
15. K. Liang, L. Fang, D. S. Wong, and W. Susilo, "A ciphertext-policy attribute-based proxy re-encryption with chosen-ciphertext security," in *Proceedings of the 5th IEEE International Conference on Intelligent Networking and Collaborative Systems*, 2013, pp. 552-559.
16. T. Kitagawa, H. Kojima, N. Attrapadung, and H. Imai, "Efficient and fully secure forward secure ciphertext-policy attribute-based encryption," in *Proceedings of Information Security*, 2015, pp. 87-99.
17. C. Chen, Z. F. Zhang, and D. G. Feng, "Efficient ciphertext policy attribute-based encryption with constant size ciphertext and constant computation-cost," in *Proceedings of International Conference on Provable Security*, 2011, pp. 84-101.
18. Z. Zhou and D. Huang, "On efficient ciphertext-policy attribute based encryption and broadcast encryption," in *Proceedings of the 17th ACM Conference on Computer and Communications Security*, 2010, pp. 753-755.
19. Z. Zhou, D. Huang, and Z. Wang, "Efficient privacy-preserving ciphertext-policy attribute based-encryption and broadcast encryption," *IEEE Transactions on Computer*, Vol. 64, 2015, pp. 126-138.
20. Q. Li and F. Zhang, "A fully secure attribute based broadcast encryption scheme," *International Journal of Network Security*, Vol. 17, 2015, pp. 255-263.
21. T. Kitagawa, H. Kojima, N. Attrapadung, and H. Imai, "Efficient and fully secure forward secure ciphertext-policy attribute-based encryption," *Information Security*, Springer International Publishing, 2015, pp. 87-99.
22. A. Ge, R. Zhang, C. Chen, C. Ma, and Z. Zhang, "Threshold ciphertext policy attribute-based encryption with constant size ciphertexts," in *Proceedings of Australasian Conference on Information Security and Privacy*, 2012, pp. 336-349.
23. X. Fu, L. Fagen, and S. Zeng, "Oblivious transfer with fine grained access control from ciphertext policy attribute based encryption in the standard model," *International Journal of Future Generation Communication and Networking*, Vol. 9, 2016, pp. 285-302.
24. K. Liang, L. Fang, W. Susilo, and D. S. Wong, "A ciphertext-policy attribute-based proxy re-encryption with chosen-ciphertext security," in *Proceedings of the 5th IEEE International Conference on Intelligent Networking and Collaborative Systems*, 2013, pp. 552-559.
25. J. H. Park and D. H. Lee, "Anonymous HIBE: Compact construction over prime-order groups," in *Proceedings of IEEE Transactions on Information Theory*, Vol. 59, 2013, pp. 2531-2541.
26. R. Bobba, H. Khurana, and M. Prabhakaran, "Attribute-sets: A practically motivated enhancement to attribute-based encryption," in *Proceedings of European Symposium on Research in Computer Security*, 2009, pp. 587-604.

27. C. C. Lee, P.-S. Chung, and M.-S. Hwang, "A survey on attribute-based encryption schemes of access control in cloud environments," *International Journal of Network Security*, Vol. 15, 2013, pp. 231-240.
28. J. A. Akinyele, M. Green, and A. Rubin, "Charm: a framework for rapidly prototyping cryptosystems," *Journal of Cryptographic Engineering*, Vol. 3, 2013, pp. 111-128.

P. G. Shynu received his ME degree in Computer Science and Engineering from College of Engineering, Anna University, Chennai, India. He is currently pursuing his Ph.D. degree in the School of Information Technology and Engineering, VIT University, Vellore, India. His research interests include cloud security and privacy, ad-hoc networks and big data.

K. John Singh received his Ph.D. degree in the Faculty of Information and Communication Engineering from Anna University, Chennai, India in 2013. He received MS degree in Information Technology from Manonmaniam Sundaranar University, Tirunelveli, India in 2002 and M.Tech degree in Computer and Information Technology from Center for Information Technology and Engineering of Manonmaniam Sundaranar University, Tirunelveli, India in 2004. Currently, he is working as Associate Professor in the School of Information Technology and Engineering, VIT University, Vellore, India. His research interests include network and information security, cloud security and image processing.