INTERNATIONAL JOURNAL FOR TRENDS IN ENGINEERING & TECHNOLOGY
VOLUME 3 ISSUE 3 – MARCH 2015 – ISSN: 2349 – 9303

Attribute Based Encryption with Attribute Hiding in Cloud Storage

D.VADUGANATHAN[1]
Angel College of Engineering and Technology
[1]Anna University, CSE,
Vaduganathan.kce@gmail.com

S.RAMASAMI[2]
Angel College of Engineering and Technology
[2]Anna University, CSE,
sramasami@gmail.com

Abstract— Cloud computing is widely used technologies that provides on-demand self services. One of the main services is cloud storage. Cloud storage is used to store up and access the data anywhere from the cloud. To have a have power over the data we necessitate a fine-grained access control system. One of the Fine-grained access control system is Attribute Based Encryption (ABE). The ABE system is used to provide access control using attributes. Attributes may be anything. For example, it may person's roles or date. In ABE system the encryption phase consists of access rules. Rules are depended on the data owner and the consumer. It is not hidden in the existing systems. Through this, users can get the useful information from access rules without decryption. To improve the effectiveness of the ABE system, a new ABE algorithm is proposed. The proposed ABE algorithm has the features hiding Attributes and Access Policies. Through this, users getting useful information from the cipher text are avoided. Attributes and Access policies are hidden by the hash function and polynomial functions. The proposed ABE system ensures data privacy and policy privacy. Through this, the efficiency of the attribute based encryption system is improved.

Index Terms— cloud computing, data sharing, cloud storage, ABE, Access policies, Secret sharing

1 INTRODUCTION

CLOUD COMPUTING Cloud computing is mainly used in IT industry. It provides many on-demands self services. The services are categorized into public and private. The private services are used by only the people who are all authorized by the organization. So that private services are always secured. That is, data can access by organization's users alone. But in the case of open services lot of users will access the services. The most important challenges in open service is the secrecy of the data. Cloud storage is one of the public services. So that it concentrate on the cloud storage's security. Cloud storage requires the fine-grained access control. Cloud storage is used to store up and access the information in the cloud computing. Cloud storage comes under the public services, so it cannot have a controllable sharing of data. To ensure the controllable, secure sharing in the data, use the encryption and decryption mechanisms. These encryption and decryption mechanisms used to share the data securely. There are many encryption and decryption systems. But public-key and secret key cryptography systems are previously used mechanisms. These are used only when the identity of users is fixed. But these are not used for dynamic users. To avoid this type of problem the concept of Attribute Based Encryption systems is proposed. The ABE system is a predicate based encryption. It defines the encryption part with Access Policies. Access policies are defined by the Access tree. Access tree is created by the set of Boolean formulas. Boolean formulas consists of set of AND, OR. There are many types in the Access tree. One types of access tree is threshold-access tree. Each non-leaf nodes of access tree are defined by the threshold values. the leaf-nodes of Access tree is associated with the attributes. If the thresholds of the non-leaf nodes are satisfied, the n the leaf nodes of the attributes can access the data else cannot access the data. This type of access tree with threshold is defined as threshold-access tree. This access trees uses the Linear Secret Sharing Scheme (LSSS) matrix to generate the keys for attributes. these Access trees are associated with the cipher text. ABE system generates the keys associate with the set of attributes. If these attributes are matched with the cipher text, the n the decryption can be done else the decryption cannot be done. It is called as the attribute based encryption system. It is also called as the predicate based encryption. Attribute based encryption is the efficient and reliable method for the data sharing in cloud computing. Hiding the data alone is not fully secured in the ABE system. Because ABE system's cipher text is associated with Access policies. The access policy consists of information that contains the attributes and access formulas of the consumer. Access Policies also important in the ABE system. Hence it should be heeded. If these Access policies are revealed, the n there is the chance for breaking the cipher text easily. The access policy revealing problem will overcome by many techniques. Here main thing is to hide the Access policies in the cipher text.

This proposed scheme is used to hide the access policies and attributes. That is users cannot get any information about the access policies and attributes. Some of the Literature surveys say that predicate encryption is efficient for hiding the attributes and access policies in the encryption. Predicate based encryption (PBE) is defined as the predicates which are associated with the secret keys and the cipher text. If the predicates are matched with the cipher text predicates, the n the decryption can be done else decryption cannot be done. But from the other parameter point of view, it has many disadvantages. Here, the main challenge of ABE system is to provide the ABE algorithm that should hide the access policies and attributes also the algorithm should be efficient in the other parameter point of view. Other parameters are scalability, effectiveness and efficiency.

1.1 Contribution of paper

In this paper, the ABE system which is secure in the form of data and also as the policies and attributes has been proposed. From the other parameter point of view ABE is efficient and scalable. In the proposed ABE system users cannot get any information about access policies and attributes. Here the proposed system is ensuring the secrecy of data, attributes and access policies.

1.2 Organization of paper

The other parts of the paper are organized as below: Section 2 describes the Literature survey. Section 3 describes the cloud computing architecture and security requirements. Section 4 describes prerequisites of cryptography and general framework of attribute based encryption system. Section 5 describes the proposed concepts of ABE system. Section 6 analyzes the security of proposed ABE system. Section 7 consists conclusion of the paper.

3 RELATED WORKS

To avoid the disadvantages of identity based encryption (IBE), the Attribute Based Encryption system is introduced. The ABE system is used to provide the fine-grained access control systems. Identity based encryption system is defined based on the identity of the encryption and decryption mechanisms. Identity may be a mobile number or e-mail id of the persons. These identities are static. IBE is not efficient for dynamic or large scale organization. To avoid this, the concept of ABE system is introduced. Attribute based encryption can be done by the set of attributes. The cipher text is associated with the access policies. Secret keys are associated with the set of attributes. The cipher text can decrypt only when the access policies attributes and key attributes are satisfied. Otherwise decryption of the cipher text cannot be done. But it has a disadvantage that access policies and attributes are revealed.

To avoid this type of revealing of access policies and attributes, the concept of functional encryption (FE) is introduced. Through the FE it is easier to hide the access policy and attributes. Functional encryption consists of access structures with secret keys. While generating the secret keys, the access policies are associated with the keys. Policies and attributes are hidden here. However, access structure is defined by the authority. Data owner should believe the authority alone [7]. Cloud mask is used to hide the attributes and policies. Cloud mask uses three roles. They are data managers, storage service, users. Data manager is responsible for doing encryption with access policy. Storage service is used to store up the documents. Users can access the data from the storage service. Here through data managers, it is easier to hide the attributes and access policies. However, it is difficult for data managers maintain dynamic access policies [8]. Predicate based encryption is one of the techniques used to hide the access policies and attributes. the predicate based encryption is same as attribute based encryption except that here attributes and access policies are hidden. PBE consists of the predicates, i.e. set of access rules. If the access rules are satisfied with the keys the n the decryption can be performed else we cannot do the decryption. If access rules are satisfied with the keys the n partial transformation is performed in the cloud server. So that the re is a chance for cloud server to learn useful information about the access policies that are gathered from the cipher text. PBE is efficient to hide the access policies and attributes. However, in the case of partial decryption cloud servers can learn the useful information [13].Blind extraction is one the policies and attributes hiding method. This blind extraction is used in the database cipher text search. While doing search on the database there is a chance for getting useful information. To avoid this type of search secrecy problem, use the blind extraction method. Using the blind extraction method, an efficient search on the database chipper texts can be done. Attribute based encryption and predicate based encryptions are not sufficient for the database cipher text search. However, authority is responsible for doing searches with query responses. So that single point failure is occurring in the blind extraction. It is difficult to maintain the blind extraction in the cipher text searches [5]. Hierarchical predicate based encryption (H-PBE) is one of the encryption techniques used to improve the scalability of the predicate based encryption. Scalability is not sufficient in the predicate based encryption. So that hierarchical predicate based encryption is introduced. Hierarchical predicate based encryption consists of different roles. They are global authority and local authority. Global authority is responsible for managing all the local authorities. Each local authority responsible for maintaining the set of attributes. Here the secret keys are generated hierarchically. Hierarchical predicate based encryption system uses the polynomial function to produce the keys. However, maintaining all local authority by global authority is difficult. There is a chance of single point of failure may occur in global authority [11]. Next introduces the privacy preserving attribute based encryption (PP-ABE). PP-ABE consists attributes in the two forms. One form is application level attributes and the second form is algorithm level attributes. Application level attributes are the roles of the human. For each application level attributes, there is one form of algorithm level attributes is maintained. Algorithm level attributes are saying the positive and negative occurrences of the application level attributes. PP-ABE algorithm doing the mapping between application levels attributes and algorithm level attributes. Algorithm level attribute are visible by the users. Even though the algorithm level attributes are visible users cannot learn anything from this attributes. Application level attributes are not visible by the users. Through this PP-ABE algorithm can easily hide the attributes and policies. Even though the PP-ABE algorithm hiding attributes and polices from the users and cloud server, to maintain both application level attributes and algorithm level attributes is difficult. Conflicts occurring between the application level attributes and the algorithm level attributes. these attributes are maintained only by the AND logics [12]. Next introduces the Secret sharing- attribute based encryption (SS-ABE) to hide the access policies and attributes. SS-ABE is used to avoid the keys, transforming to unauthorized persons. SS-ABE scheme uses KP-ABE systems. KP-ABE system is defined as the key policy-attribute based encryption. Key policy attribute based encryption is defined as the secret keys are associated with the set of access rules. Access rules are not associated with the cipher text. So that Access policy secrecy problem will not be occurring in the KP-ABE systems. Through Key policy attribute based encryption systems can hide the access policies and attributes. However, KP-ABE systems do not have good scalability. Improving scalability in the KP-ABE system is difficult one. Authority is only responsible for maintaining access rules with keys. So there is a chance of single point of failure may occur in the KP-ABE systems. Next introduces the predicate based encryption for inner products. IPV is defined as the inner products predicate based encryption. IPV access rules are defined by the Boolean formulas. These Boolean formulas can be expressed as the two formats. That is, the conjunctive normal form and the disjunctive normal form. These Boolean formulas are not sufficient comparatively with the attribute based encryption. These above literature survey shows that there is no effective method for hiding the attributes and access policies in the cloud computing. Because each method has some of the disadvantages. These are the methods for managing the data secrecy and access policy secrecy in the attribute based encryption system. Access policy revealing occur maximum in the cipher text- attribute based encryptions. It is one types of attribute based encryption system. Key-policy attribute based encryption system is another type of attribute based encryption system. Here access policy secrecy and attributes secrecy problem do not occur. Because authority is responsible for producing the secret keys with access rules. Here the access policy is associated with the secret keys. So that policy secrecy problem is avoided.

4 Cloud Computing Architecture

Cloud computing is used to provide the on-demand self services based on the pay of use. Such that users should pay based on the services what the y are receiving. These services are classified in cloud computing by three bases. They are platform as a service, infrastructure as a service, and software as a service. Platform as a service is defined as the service provided by the cloud computing that fully depends on the platform. For example, database queries need the platform like SQL. This type of SQL platforms can be used as a service. No need to install the SQL in our computer. This platform will be directly used from the internet. It is called as the platform as a service (PaaS). It is one of the layers of cloud computing architecture. Software as a service is the part of the cloud computing layers. Google docs are the one of the examples of software as a service. Google docs are used to access the Microsoft documents without installing the Microsoft software in our own computer. It is one of the main services in the cloud computing architecture that is used in our day to day life. Infrastructure as a service is used to provide storage services in the cloud computing. Dynamic applications need more scalability requirements. It is the most important challenges in our cloud computing. To manage these challenges in cloud, on-demand storage service is required. This type of on-demand storage is provided by the infrastructure as a service. The above three type of services are provided by the internet. The internet is also called as a cloud carrier. Such that internet is used to pass the services to all.

4.1 Cloud storage

Cloud storage is one of the services in the cloud computing. Cloud storage is mainly used for storing the document which is more than 25GB. Because these types of documents cannot be sent by the people. So that it will be stored in a common place. From that place, people will upload the it data. That is called as the cloud storage. Users will access the data from the cloud storage. The main challenge in the cloud storage is unavailability of controllable sharing. Cloud storage will be accessed by the internet. the data can access from anywhere in the world. This is the main advantage of cloud computing. Cloud storage act as a common storage device in the cloud computing for sharing the data

4.2 Security requirements

Data confidentiality is the used to ensure that the authorized persons only can access the data. Unauthorized persons cannot satisfy the attributes with access policies. Data confidentiality should satisfy the attribute based encryption.

Attribute based encryption should satisfy policy secrecy and attribute secrecy, such that policy and attributes should be hidden from users. So that it is quite easier to ensure full security in the attribute based encryption system.

Collusion resistance is defined as the users cannot decrypt the cipher text by combining the different attributes in the attribute based encryption. Collusion means that even the single attributes cannot satisfy the access policy. Satisfaction of access policy is done by combing the different attributes. This type of collusion should be avoided from the attribute based encryption.

5 Prerequisite Of Cryptography

To define the attribute based encryption the background of cryptography is necessary.

4.1 Access tree

Access structure or access tree is defined as the collection attributes A. Subset of attributes A is defined as the authorized attributes. Otherwise, it is called as the unauthorized sets.

4.2 Bilinear pairing

G1 and G2 is the cyclic group. e:G1xG1→G2 if $e(g^a,g^b)=e(g,g)^{ab}$. g denotes the generator of the group G1.

4.3 Bilinear Diffie-hellman

Bilinear diffie-hellman algorithm is used to compute the $e(g,g)^{ab}$ € G2 by using the generators of the cyclic group G1. Main advantages BDH algorithm is used to compute the bilinear map €(k), where k is the security parameter.

4.4 One-way Anonymous protocol

One way anonymous is defined as without the interaction between two parties, the y can share the ir session keys in the form of anonymous manner. In two participants interactions one participant is in anonymous manner and another participants is in non-anonymous manner. It is called as the one way anonymous key agreement. For example, Alice has the private key, selects the random number and finds the session key. the session key is forward to bob. the n bob will find the session key using his private keys. It is called as the one-way anonymous key agreement.

4.5 General framework for ABE

Attribute based encryption system consists of following polynomial time algorithms.

Setup→param(PK,MK). The key generation center generates the public key and private keys.

Keygen(set of attributes, secret key) → secret key for each user. In the se key generation, a set of attributes and secret keys are taken as the inputs and produces the output as secret keys for each user.

Encrypt(public_key,master_public_key,message, Access_policies)→ cipher text. In encryption ABE takes public key,master public key,message and access polices are taken as the input and produces the output as cipher text.

Decrypt(Cipher text,Secret key)→Message. In this phase cipher text and secret keys are taken as the input, produces the output as original message

5 Proposed Scheme

The above literature survey shows that most ABE algorithms can hide the access policies but the y are failed to satisfy the efficiency or expressiveness of the ABE algorithm. To improve ABE algorithm efficiency as well as ensuring the data secrecy and the policy secrecy, following steps is done in the proposed algorithm.

5.1 Access tree

Here consider the Access tree as P. It consists of nodes that represent the threshold. Each node ‘u’ has the number of children and is denoted as the ‘n’. The node ‘u’ has the threshold value ‘l’. Threshold value ‘l’ lies between 0 and values of number of children. More formally, a threshold of each node is defined as (0< l< n).

Access tree consists of leaf-nodes that denote the attributes in the access policies. Each nodes has the parent nodes that is denoted by par(node). For example, parent node v in access tree can be represented by par(v). Parent’s child can be identified by the integer numbers. Child nodes integer number is denoted by the values as index(v). The index(v) is used to provide the unique identification to the child nodes.

5.2 Access tree satisfaction

Access tree is represented as the ‘A’. The subtree of access tree is represented by A_v at node ‘v’. Set of attributes Attr is satisfied with the node v and it can be represented as A_v(Attr)=1. It can be repeated in each subset of access tree. The access tree A_v returns ‘1’ when the threshold value ‘l’ is satisfied. It is called as the Access tree satisfaction. It is used to guarantee that the access policies are satisfied in the attribute based encryption.

5.3 Proposed scheme construction

Bilinear mapping is used in the attribute based encryption. G1 is defined as the bilinear mapping if it is satisfying G1 x G1 → G2. Secret sharing method is used in the access tree. Secret sharing is defined as the access tree consisting of secret values S. This secret values should be shared by the different attributes. The secret values are shared by the polynomial value (P). For example, jth attributes gets the share that can be represented by (j,P(j)). These shares can be represented by the polynomial values like P(Y1)…….P(Y10). At fist the secret keys can be represented by P(0)=S. these polynomial values can be identified by Lagrange interpolation. The Lagrange polynomial values can be identified by,

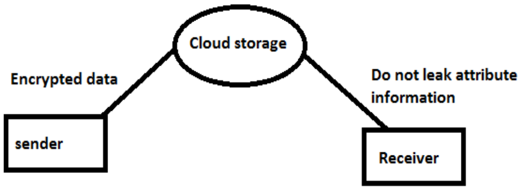
$$P(0) = \sum \lambda_j P(Y_j)$$

Where λ is the Lagrange coefficients and it is represent as

$$\lambda_j = \prod \frac{X_j}{X_j - X_i}$$

Here i,j represents indexes of attributes.

5.4 System Architecture



5.5 Setting up private and public keys

KGC (key generation center) is involved in setting up phase. KGC first chooses group G1 and the hash function. KGC is used to produce the public key and private keys. The hash function can be represented as the H:{0,1}→G1. α,β is selected by the random numbers. These random numbers can be represented by KDC, u be the generator of the group.

α = random();
β = random();
public key = {G1,u,h=u^{β},e(u,u)$^{\alpha}$ }
private key = {β,u^{α} }

Above public key and private keys will return from the setup phase. Setup algorithm can be represented by,

```
Setup {
        KDCrandom() = α;
        KDCrandom() = α;
        PubK = {G1,u,h=$u^{\beta}$,e(u,u)$^{\alpha}$ }
        MaterK = {β,$u^{\alpha}$ }
        Return (PubK, MaterK);
    }
```

5.6 Key generation for attributes

KDC is used in this phase. Here the KDC is responsible for generating the keys for each attribute as well as attribute keys should be personalized with the particular user. When a new user is arrived, KDC will select the random number for that particular user. Using that random number for each user, α, β produce the AK random numbers. the n KDC produces the attribute keys for each attribute in the set A. Here A represents the group of attributes. Through the user key and attribute keys the personalized user’s attribute keys are produced. These user key and attribute keys will be given to the user. The above procedure in the Key generation can be represented as a summarized algorithm form,

```
KeyGeneartion (MasterKey,SetofAttributes,Usersid) {

Userrandom = H(Userid)

UseKey.MasterKey = $u^{\frac{\alpha+userrandom}{\beta}}$

If j in A the n

        $Userrandom_j$  =  KDCranatt()

        $UseKey.MasterKey_{ju}$ = $u^{Userrandom}$.H(j)$^{Userrandom_j}$

Endif

Retrun { UseKey.MasterKey, $UseKey.MasterKey_{ju}$ }
```

.

Notations	Descriptions
Userrandom	Random key for each user in ABE
UseKey.MasterKey	Using random key and masters keys, random number is produced
$Userrandom_j$	Keys for each attribute in ABE
$UseKey.MasterKey_{ju}$	Personalized key

Theabove Key generation algorithm shows that KDC takes Master keys and set of attributes to produce the user personalized attribute key.

5.7 Encryption

Users before putting their data in cloud have to encrypt the message with access policies. Access policies cannot be viewed by the users and cloud. Through this can ensure the data privacy and access policies privacy. Data privacy is given by the encryption. Secrecy of access policies is ensured by polynomial values is assigned to the each node in the tree. Access tree consists leaf nodes, which contains information about the attributes in the access policies. It is not secure if it is not replaced by the polynomial values. In access tree for each leaf node polynomial values O_j computed by following,

$$O_j = e((u^{\beta})^{\alpha}, H(\text{Plain attribute}))$$

Where a is random number selected by the data owner. Through this can hide the attributes.

Access tree each node consists set the polynomial values in the following manner. In access tree first consider the root node, which has the polynomial value by $P_{root}(0) = O$. Each node has the degree which is defined as the threshold value-1. That means the degree of the node is less than the threshold value. Other than the root node for example, node 'c' of the polynomial constant this is represented by following,

$$P_c(0) = P_{par(c)}(\text{index}(c))$$

Have to repeatedly define the Polynomial values. The above encryption is summarized in the following,

```
ENCRYPTION {
            $Cipher_1 = h^O$
            $Cipher_2 = Message.e(u,u)^{\alpha O}$
Start
            For each leaf node k find the polynomial values
Do
            $Cipher_k = u^{Pk(0)}$
            $Cipher^1_y = H(i)^{Pk(0)}$
Where k is the leaf node.

Finish
    Return {Cipher1,Cipher2,$Cipher_k$, $Cipher^1_y$}
        }
```

The encryption algorithm shows that Access tree (A), Public Key, Access Policies as the input and produces the output as Cipher text. In the encryption shows that plain attributes is hide by the polynomial values. Through this cloud cannot learn any useful information from the cipher text.

Notations	**Descriptions**
Oj	Polynomial value
U	Generator of group
Cipher1	Message Encryption
Cipher2	Message Encryption
$Cipher_k$	Each leaf node's att are hidden
$Cipher^1_y$	Attributes are replace by hash function with polynomial

5.8 Decryption

In the attribute based encryption, decryption can done in the form two steps. In first step, decryption is done in the access tree node level. This node level decryption is done through the secret keys of users. Second decryption is data decryption. This second decryption is done by using keys gathered from node level decryption keys. For example, consider a set of attributes {PHD-student,Angel-college,Book-x,Book-y,Book-z}, access policies can be defined by the following diagram.

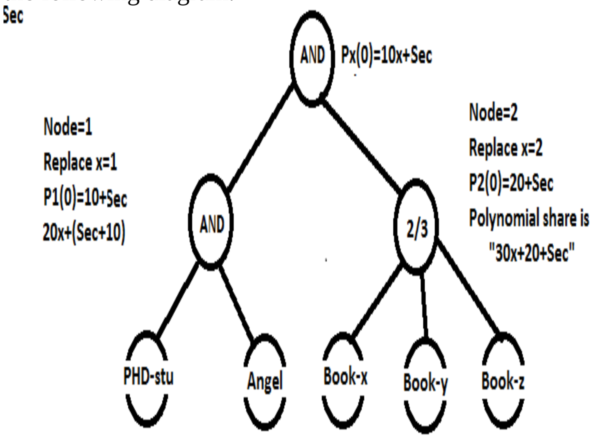


Figure.1

Decryption at Node1 level,

F1 = e(Aggregation of attributes keys, $Cipher_k$)

$$\left\{ u^{\frac{r}{(\gamma+P)}}, u^{\frac{r}{\gamma+A}} . \right\}$$

It represents aggregation of node 1 level. *P* denotes the PHD-Student attribute. *A* denotes the Angel college. It is represented as N1.

Node1 cipher text is represented by the following,

$$Cipher\_node1 = h^{(Sec+10).\alpha(\gamma+P)(\gamma+A)}$$

$$F1 = e(u,h)^{r(Sec+10).\alpha}$$

F1 represents that some field can share at node 1 attributes level. Use the same way to find field share at node 2 attributes.

Decryption at Node2 level,

Field share at node 2 can be represented by following,

$$F2 = e(u,h)^{r\alpha(Sec+20)}$$

Combination field share is defined for Access tree by following,

$$F_x = \prod F^{\Delta_j Sec_y}$$

Here , $\Delta_j Sec_y$ represents Lagrange coefficients.

$$\text{Fx} = F_1^{\Delta_j Sec_y} F_2^{\Delta_j Sec_y}$$

$$= e(u,h)^{r(Sec+10).\alpha} e(u,h)^{r\alpha(Sec+20)}$$

$$\text{Fx} = e(u,h)^{r\alpha Sec}$$

F_x represents decryption at the node level. It is not completely decrypted. So decrypt it in the Data level using the above field share F_x .

Data Decryption Level,

Data decryption level can be done through the Fx field share.

$$= \frac{e(h^{Sec}, u^{(\alpha+r)/\beta})}{(e(u,h)^{r\alpha Sec})^{1/\alpha}}$$

$$N \qquad = e(u,h)^{rSec}$$

From cipher text, $Cipher_2$ can be represented as follows,

$$Cipher_2 = Message.e\,(u,h)^{rSec}$$

From the cipher text we can decrypt the data through the value of 'B', by following equations,

$$= \frac{Cipher_2}{N}$$

$$= \frac{Message.e\,(u,h)^{rSec}}{e(u,h)^{rSec}}$$

$$= Message$$

The above equation shows that decryption can be performed partially in the node level. After that data is fully decrypted in the data level decryption.

The above decryption shows that the steps and procedures are followed in the attribute based decryption using access tree. We can summarize the above decryption procedure by following,

```
PartialDecryption {
        While consider each leaf nodes
        Do
        Checks attributes secret keys is satisfy or not by,
        e(Cipher,Attributessharekeys)
        if(Attributes keys satisfied)
        {
              Return B = $e(u,u)^{rSec}$
        }
      Else
      {
              Return 0;
      }
DataDecryption{

        = $\frac{cipher}{e(h^{Sec}, u^{(\alpha+r)/\beta}) /(B)^{1/\alpha}}$
      Return Message
        }
```

The above algorithm shows that partial decryption checks whether the attribute key is satisfied with the access policies or not. If attribute keys are satisfied, the n it returns the partial decryption of the node and it produces the data decryption

NOTATIONS	DESCRIPTION
A	Access tree
P	PHD-Student attribute
A	Angel College attribute
γ	Random component
F	Field share of attributes
Sec	Random Polynomial value
α , β	Random component
B	Checks attributes satisfaction
U	Group generator
F1	Access tree's field share for attributes PHD and Angel
F2	Field share of node2 in A

6 SCHEME IMPLEMENTATION

Access policies hiding attribute based encryption algorithm is implemented by cpabe-toolkit. It is one of the toolkit used for implementing the attribute based encryption. the working of this toolkit is based on the PBC library.PBC library is defined as the pair wise cryptography library and is implemented in the Linux environment. Cpabe-toolkit is working under the GNU library. Here, the attribute based encryption with policy hiding in the C language is implemented.

7 SCHEME SECURITY

In this paper three security requirements are considered that are data secrecy, policy secrecy and collusion resistance.

7.1 Data secrecy

In this proposed scheme data secrecy is ensured. Data secrecy is also called as the data confidentiality. Data secrecy is defined as unauthorized users cannot access the data. It is called as the data secrecy. In proposed scheme partial decryption and data decryption is ensured. This partial decryption is done by user attribute keys. User's attribute keys are satisfied only by the data decryption. Otherwise, one cannot proceed with the wrong attributes. This attributes checking is done by node level in the access tree. the node level satisfaction is checked by the polynomial values. Wrong attribute users such as unauthorized users cannot perform the partial decryption. Through this data secrecy is ensured in the proposing system.

7.2 Policy secrecy

Policy secrecy is defined as the ensuring policies are not learned by cloud and outside users. This policy secrecy is ensuring in the proposed scheme. Plain attributes A_j in the leaf nodes with H(Oj).

Where $O_j = e((u^{\beta})^b, H(A_j))$

Without knowing the correct attribute key, outside users cannot compute the O_j values. Users cannot learn any information about the attribute from the hidden attribute $e((u^{\beta})^b, H(A_j))$. This attribute is computed only when we know the value $H(A_j)^{\beta}$. In the same way cloud server also cannot learn anything from the hidden attribute $e((u^{\beta})^b, H(A_j))$. Authorized users only have the values of $H(A_j)^{\beta}$. Through this, $H(A_j)^{\beta}$ value can learn the attributes from hidden attribute $e((u^{\beta})^b, H(A_j))$. So that our proposed scheme is used to ensure the policy secrecy. That is policy secrecy is ensuring from the cloud server as well as the from the outside users. For each node, create the polynomial values in the access tree. Through this it is easier to ensure the policy secrecy. Such that these leaf node's polynomial values can be find only when the user's parts of the key have contain information about the each attribute random values using $H(j)^{attributerandomkey}$. This is used to ensure the policy secrecy in the attribute based encryption system which is efficient compared with other hiding schemes.

7.2 Collusion resistance

Collusion resistance is ensured in the proposed scheme by using random values of the keys for each personalized users. Collusion is defined as, while combining the different keys of users, unauthorized persons can decrypt the original message that is called as the collusion. This is avoided in the proposed scheme by ensuring the random values of keys for each attributes. Attacker have to discover the values of $e(u,u)^{\alpha Sec}$ in the attribute based system. To find this $e(u,u)^{\alpha Sec,}$ attacker have to discover the values of

$$\frac{e(h^{Sec}, u^{(\alpha+r)/\beta})}{(e(u,h)^{r\alpha Sec})^{1/\alpha}}$$

is required. But attacker cannot find these values because to find this value, the attacker should know the attribute information. But our proposed algorithm ensures the Policy secrecy. Through this attacker cannot find the any information regarding the access policies and attributes. In this same way, the collusion resistance property in our proposed algorithm is ensured.

CONCLUSION

The proposed scheme shows that ensuring the three requirements of security is an important property of the security requirements in data secrecy. Data secrecy is ensured with the attribute based encryption. Our proposed scheme not only ensures the data polices, because ABE system defines the access policies. So that it gives security for data alone is not a fully secure ABE system. To improve this, ensure the access policy. Access policy revealing is used to provide some learning methods about the attributes and cipher text to unauthorized and cloud servers which need security other than the data secrecy. That is called as the policy secrecy. This most important security in the attribute based encryption is policy secrecy. the proposed scheme is ensuring the policy secrecy by finding the polynomial values for each attributes. This is a random value. Only authorized attributes keys have randomized key for each attributes. the se randomized keys only can find the decryption in the policy level. Through the randomized polynomial values can avoid the wrong attributes. It ensures the policy secrecy in the attribute based encryption. Final important security property is collusion resistance. Collusion resistance is ensured in our proposed algorithm by using randomized keys for each attributes.

REFERENCES

[1] Junbeom Hur, "Attribute based secure data sharing with hidden policies in smart grid," *IEEE Trans.Parallel and Distributed systems*, vol.24, pp. 2171-2180, Nov 2013.

[2] SushmitaRuj,Milos Stojmenovic,Amiya Nayak, Jia Mo, "Decentralized Access Control with Anonymous Authentication of data stored in clouds," *IEEE Trans.Parallel and Distributed systems*, vol.24,pp. 384-394, 2014.

[3] Shucheng Yu,cong wang,kui ren,wenjing lou,"Attribute Based data sharing with attribute revocation ,"*Proceedings of the 5th ACM symposium on information*, pp. 261-270, 2010.

[4] Yao Zheng,Ming Li,Shucheng Yu,Kui Ren,Wenjing Lou, "Scalable and Secure sharing of Personal health records in cloud computing using attribute based encryption," *IEEE Trans.Parallel and Distributed systems*, pp. 131-143, 2013.

[5] Yanbin Lu and Gene Tsudik, "Enhancing Data Privacy in the Cloud,"*IFIP Advances in information and communication technology*,vol.358,pp.117-132,2011.

[6] Taeho Jung, Xiang-Yang Li, Zhiguo Wan Meng Wan,"Privacy Preserving Cloud data Access with Multi-Authorities," *IEEE Trans. INFOCOM IEEE proceedings*, pp. 2625-2633, April 2013,

[7] Dan Boneh,and Amit Sahai and Brent Waters, "Functional Encryption: Definitions and Challenges," *TCC*, pp. 253-273, 2011.

[8] Hongjiao LI, Shan WANG, Xiuxia TIAN, Weimin WEI, Chaochao SUN,Daming LIU, "A Survey of Privacy-preserving Access Control in Cloud Computing,"*JCIS* pp. 5829-5846,Jul,2014.

[9] Dongyang Xu,Fengying luo,Lin Gao and Zhi Tang, "Fine-grained document sharing using attribute-based encryption in cloud servers," *INTECH*,pp.65-70,2013

[10] Ming Li,"Authorized Private Keyword Search over Encrypted Data in Cloud Computing," *International conference on distributed computing systems*.pp.383-392,2011.

[11] Shucheng Yu, "Data Sharing on Untrusted Storage with Attribute-Based Encryption," *the sis of Worcester polytechnic instuite*, .2010.

[12] Fugeng Zeng, "Predicate Encryption for Inner Product in Cloud Computing," *IJACT*, vol. 4, no. 13, pp. 52-61,2012.