Various Attribute Based Encryption Algorithms in Clouds-A Survey

P. Sivakumar1, K. Devi2 M. Navinkumar3
1,2,3Department of Computer Science & Engineering
Valliammai Engineering College, Chennai, Tamilnadu-603203

Abstract— Attribute Based Encryption method (ABE) ensures that authorized access to the data. Due to the huge user base and often unknown identities of target recipients the User based access control models are too restrictive. The ABE is highly efficient and flexible since it allows a content provider to specify an access policy and encrypt multiple messages within one ciphertext. In particular, only the owners whose attributes satisfy the access policy can decrypt the ciphertext. The ABE method provides security for appropriate users when accessing the data’s and the encryptions and decryptions are done by the user based access control attributes. The ABE authority will assign the access rights for various users. Attribute-Based Encryption is a promising cryptographic primitive which greatly enhances the versatility of access control mechanisms. The computational complexities of ABE key-issuing and decryption are getting prohibitively high because of the high expressiveness of ABE policies. This paper focuses on various concepts for ABE encryption and decryption methods for providing Security.

Key words: ABE, Encryption, Decryption, Cipher text, Cloud, Access Control

I. INTRODUCTION

Attribute-based Encryption (ABE) is one of the most suitable technologies for data access control in cloud storage as it gives data owners more direct control on access policies. Access control is the fundamental security mechanism that helps information sharing in a controllable manner. So based on the permission relationship between user attributes and resource attributes, it exerts control over which user can access which resource. In ABE system, users’ private keys are labeled with sets of descriptive attributes and ciphertexts are labeled with sets of access policies. A particular key can decrypt a particular ciphertext only if associated attributes and policy are matched. For example the Personal health records contains the following fields: name, gender, address of the patient, nurses, doctors, clinical history, drugs administered, family history of diseases, insurance policies, hospitals visited for treatment. [1] The access policies will be defined by the data owner and encryption will be carried out based on those policies. A secret key reflecting its attributes will be given to each user and the user can decrypt the data only when its attributes satisfies those access policies. An authority will be responsible for providing the secret keys for the attributes of each user. Any subset of attribute set can be used by the user for signing the messages with the secret key. The signature can be verified against a policy of attributes and verification will be successful only if the attributes used during signing satisfies the policies. The requirement of ABE with outsourced decryption is verifiability. In other words, verifiability guarantees that a user can efficiently check if the transformation is done correctly [4]. Attribute-based encryption is an advanced encryption technology where the privacy of receivers is protected by a set of attributes. An encrypter ensures that only the receiver who matches the restrictions on predefined attribute values associated with the ciphertext can decrypt the ciphertext. Membership management is an important issue in ABE system as the user joining, leaving, and updating may occur frequently in real situations.

Several expressive data access policies have been enforced by the Ciphertext Policy Attribute Based Encryption (CP-ABE). [2] With the given number of attributes, the Privacy Preserving Constant CP-ABE (denoted as PP-CPABE) significantly reduces the ciphertext to a constant size. Regardless of the number of attributes it also enforces hidden access policies with wildcards and incurs constant-size conjunctive headers. In CP-ABE, a ciphertext is embedded with an access control policy associated with user attributes. It can be viewed as a one-to-many public key encryption scheme and it enables a data owner to grant access to an unknown set of users.

To share outsourced data on untrusted storage server instead of trusted server in a secure way ABE system can be used. [11] De centralized attribute-based encryption (ABE) is a variant of a multiauthority ABE scheme where each authority can issue secret keys to the user independently without any cooperation and a central authority. In multiauthority ABE to avoid the collusion attack the user’s secret keys from different authorities must be tied to his Global Identifier (GID). In decentralized ABE every authority can freely join or leave the system without reinitializing the system. A data file has several operations like, Create, Read all, Delete, Modify, executable on itself, and each of them is allowed only to authorized users with different level of qualifications.

To enforce flexible attribute-based access control on scalable media Multi-message Ciphertext Policy Attribute-Based Encryption (MCP-ABE) can be used. It encrypts multiple messages within one ciphertext [12]. It constructs a key graph which matches users’ access privileges and encrypts media units with the corresponding keys and then encrypts the key graph with MCP-ABE. The data consumers with the required user attributes only can decrypt the encryption of the key graph and the media units.

II. BACKGROUND

A. Bilinear Maps:

It is essential to know about bilinear maps before discussing about various ABE schemes. It is a tool for paring based cryptosystem. Bilinear maps are called parings since it associates two vector elements with
another vector element. It is a function that combines the elements of two vector space and yields another element with third vector space.

Consider $G$ be a cyclic group of prime order $p$ generated by $g$. Let $G$ be a group of order $p$. Map is defined as $e: G \times G \rightarrow G_T$. The map satisfies the following properties:
1) $e(u, u^a) = e(u, v)^{ab}$ For all $u, v \in G$ and $a, b \in Z_p$.
2) Non-degenerate: $e(g, g) \neq 1$.
3) $e$ is efficiently computable.

B. Key Policy ABE (KP-ABE):

Key Policy ABE is an important application in data sharing on untrusted cloud storage. In this method messages can be encrypted with respect to subsets of attributes [20]. It enables senders to encrypt messages under a set of attributes and private keys associated with access structures that specify which ciphertexts which cipher texts a user (the key holder) is able to decrypt.

The four main steps involved in ABE are:
1) Setup: This algorithm sets up the groups, and public parameters.
2) Key Generation (Key Gen): This algorithm generates secret keys, depending on the set of attributes a receiver has.
3) Encryption (Encrypt): A sender encrypts the message using its public keys.
4) Decryption (Decrypt): This algorithm enables a receiver with matching attributes to decrypt the message.

C. Decentralized KP-ABE (DKP-ABE)

Decentralized key policy ABE scheme is to protect the users privacy using which each authority can issue secret key to users separately without having any idea of his GID. In Decentralized key policy ABE, without cooperation of a central authority the individual authorities can issue the secret key to user independently [15]. So there is no need to trust on to the central authority hence even if multiple authorities are corrupted they can’t collect the users attributes by tracing users GID.

A decentralized KP-ABE scheme consists of the following five algorithms:
1) Global Setup: This algorithm takes as input a security parameter $\lambda$ and outputs the system parameters $\text{params}$.

   $\text{Global Setup}(1^{\lambda}) \rightarrow \text{params}$

2) Authority Setup: Each authority $A_i$ takes the security parameter $\lambda$ as input and generates their secret-public key pair $(SK_i, PK_i)$ and an access structure $A_iGID$, for $i=1, 2, \ldots N$.

   $\text{Authority Setup}(1^{\lambda}) \rightarrow (SK_i, PK_i, A_i)$

3) Key Gen: Each authority $A_i$ takes as input his secret key $SK_i$, a global identifier GID and a set of attributes $I_iGID$, and outputs the secret keys $SK_iU$.

   $\text{KeyGen}(SK_i, GID, A_iGID) \rightarrow SK_iU$ Where $A_iGID = AGID \cap A_i$, AGID and $A_i$ denote the attributes corresponding to the GID and monitored by $A_i$, respectively.

4) Encryption: This algorithm takes as input the system parameters $\text{params}$, a set of attributes $AC$, a message $M$, and outputs of the ciphertext $CT$.

   $\text{Encryption}(\text{params}, M, AC) \rightarrow CT$

   Where $AC = [A_1^C, A_2^C, \ldots, A_N^C]$ and $A_i^C = AC \cap A_i$.

5) Decryption: This algorithm takes as input the Global Identifier GID, the secret keys $\{SK_iU\}_{i\in IC}$, the ciphertext CT, and outputs message the $M$.

   $\text{Decryption}(\text{GID}, \{SK_i^U\}_{i\in IC}, CT) = M$

Where $L_C$ is the index set of the authorities $A_i$ such that $A_i \in L_C$. Where $A_C^i \in \{\phi\}$.

D. Cipher text Policy ABE (CP-ABE):

It a user’s private-key is associated with a set of attributes and a ciphertext specifies an access policy over a defined universe of attributes within the system. A user will be able to decrypt a ciphertext only if its attributes satisfy the policy of the respective ciphertext. It is a kind of today based encryption.

CP-ABE consists of the following algorithms:
1) Setup which initializes the public key $PK$ and master secret key $MSK$ parameters.

2) Key generation (Gen), which generates the secret key $SK$ of the users using the master secret key $MSK$, and a set of attributes $S$ that describe the key.

3) Encryption (Encrypt) which encrypts the message $M$ using the public parameters $PK$ and the access policy $A$ and outputs a ciphertext CT.

4) Decryption (Decrypt) which takes as input the public parameters $PK$, a ciphertext $CT$, which contains an access policy $A$. If the set of attributes satisfy the access policy, then the decrypt algorithm returns the message $M$.

E. CP-ABE with Dynamic Membership:

Dynamic membership is always required in most applications. Since user may join, leave at any point of time and hence updating the user attributes may occur frequently in real time situations the dynamic membership is the big challenge in real time ABE [5].

1) Setup: Take a security parameter as input and generate public parameter and master key.

   $\text{Setup}(k) \rightarrow (PK, MK)$

2) Enrollment: The input is where is the set of user’s attributes and is the vector of the values corresponding to the attributes in. This algorithm will be performed when user enrolls in the system. The private key $D$ will be issued to user.

   $\text{Enrollment}(S_i) \rightarrow D$, where $S_i = (A_{i}(m_{ij}) \forall j \in A_i)$

3) Leaving : When user is revoked by the system, this algorithm will be performed and then will be updated.
Leaving (i)

4) Updating: This algorithm will be executed when user applies for modifying her/his attribute to be with value. After performing this algorithm, the updated D, will be given to user and will also be updated.

Updating \((i, j, m) \rightarrow D_i\)

5) Encryption: The inputs are the public parameter, an access tree structure \(T\), a key, \(\lambda\) and a message . This algorithm computes and encrypts under \(T\) where a symmetric encryption algorithm with input is key. Then it outputs a ciphertext.

Encryption \((PK, T, K, M) \rightarrow CT\)

6) Decryption: This algorithm takes a ciphertext and the private key \(D\) of user as input. If user’s satisfy the access tree structure in , the algorithm will decrypt the ciphertext and output the message; otherwise will be returned.

Decryption \((CT, D_i) \rightarrow M\)

F. Privacy-Preserving-CP-ABE (PP-CP-ABE):

Ciphertext Policy Attribute-Based Encryption makes use of expressive data access policies and each policy consists of a number of attributes. Privacy preserving ciphertext uses a hidden policy construction such that the recipients’ privacy is preserved efficiently [2]. Privacy preserving ciphertext ABE is more flexible since a broadcasted message can be encrypted by an expressive hidden access policy, either with or without explicit specifying the receivers and significantly reduces the storage and communication overhead.

The PP-CP-ABE scheme consists of four fundamental algorithms:

1) Setup: This algorithm takes the security parameter \(1^\lambda\) and the number of attributes in the system \(k\) as input and returns public key \(PK\) and master key \(MK\). The public key is used for encryption while the master key is used for private key generation.

Setup \((1^\lambda, k) \rightarrow (PK, MK)\)

2) Key Gen: This algorithm takes the public key \(PK\), the master key \(MK\) and the user’s attribute list \(L\) as input and gives out the private key of the user.

KeyGen \((PK, MK, L) \rightarrow Private\ Key\)

3) Encrypt: This algorithm takes the public key \(PK\), the specified access policy \(W\) and the message \(M\) as input and gives out the ciphertext CT such that only a user with attribute list satisfying the access policy can decrypt the message. The ciphertext also associates the anonymized access policy \(W\).

Encrypt \((PK, W, M) \rightarrow CT\)

4) Decrypt: This algorithm decrypts the ciphertext when the user’s attribute list satisfies the access policy by taking public key \(PK\), the private key \(SK\) of the user as input and gives out the original message \(M\), which only includes the anonymized access policy \(W\) as input.

Decrypt \((PK, SK, CT) \rightarrow M\)

G. Securely Outsourcing ABE (SO-ABE):

Outsourcing is a trend that is becoming more common in information technology. In some cases, the entire information management needs to be outsourced. Securely outsourcing ABE not only protects sensitive information by enabling computations with encrypted data, but also protects users from malicious behaviors by validating the computation result. [4] In SO-ABE, access policy methods are used for the encryption and decryption of messages. SO-ABE model provides following algorithms:

1) Setup: The setup algorithm takes security parameter as input and gives out public key \(PK\) and a Secret key \(SK\) as output.

a) Global Setup: This step takes security parameter \(L\) as input and gives out system parameters params as output.

\((1^\lambda) \rightarrow \text{params}\)

b) Authority Setup: Each authority \(Ai\) generates his secret-public key pair, and an access structure \(AAi\).

\(Ai \rightarrow (SKi, PKi, AAi)\)

2) Key Gen: The initialization step for key generation takes the access policy (or attribute set) \(I_{keys}\) and the secret key \(SK\) as input and gives out the key pair \((OK_{KGSP}, OK_{AA})\) as output.

KeyGen_{init}(I_{keys}, SK) \rightarrow (OK_{KGSP}, OK_{AA})

3) Key Gen: The delegated key generation algorithm takes \(V\) the access structure (or attribute set) \(I_{keys}\) and the key \(OK_{KGSP}\) for KGSP as input and gives out a partial transformation key \(TK_{KGSP}\) as output.

KeyGen_{out}(I_{keys}, OK_{KGSP}) \rightarrow TK_{KGSP}

4) Key Gen: The inside key generation algorithm takes the access structure (or attribute set) \(I_{keys}\) and the key \(OK_{AA}\) for attribute authority as input and gives out another partial transformation key \(TK_{AA}\) as output.

KeyGen_{in}(I_{keys}, OK_{AA}) \rightarrow TK_{AA}

5) KeyBlind: The transformation key blinding algorithm takes the transformation key \(TK = (TK_{KGSP}, TK_{AA})\) as input and gives out the private key \(SK\) and a blinded transformation key \(TK\) as output.

KeyBlind(TK) \rightarrow (SK, TK)

6) Encrypt: The method takes the input message \(M\) and an attribute set (or access structure) \(I_{keys}\) to be encrypted with and gives out the ciphertext \(CT\).

Encrypt(M, I_{keys}) \rightarrow CT

7) Decrypt: This takes the ciphertext \(CT\), blinded transformation key \(TK\) for access structure (or attribute set) \(I_{keys}\) as input and gives out the partially decrypted ciphertext \(CT_{part}\) as output.

Decrypt_{out}(CT, TK) \rightarrow CT_{part}

8) Decrypt: This takes the partially decrypted ciphertext \(CT_{part}\) and the secret key \(SK\) as input and gives the original message \(M\) as output.

Decrypt(CT_{part}, SK) \rightarrow M

H. Anonymous ABE (A-ABE):

Anonymous ABE is used in the situation where we need to encrypt data while keeping the access structure hidden in the ciphertext [3]. Anonymous privilege control scheme in Anonymous ABE is not only used to address the data privacy, but also the user identity privacy.
1) Setup: This algorithm takes only implicit inputs such as security parameters. Authorities authorities execute this algorithm to jointly compute a system-wide public parameter PK as well as an authority-wide public parameter yk, and to individually compute a master key MKk.

Setup (implicit params): → PK, MKk.

2) KeyGenerate: This algorithm enables a user to interact with every attribute authority, and obtains a private key SKu corresponding to the input attribute set Au.

Key Generate (PK, MKk, Au) → SKu.

3) Encrypt: This algorithm takes the public key PK, a message M, and a set of privilege trees \{Tp\} as input \( p \in \{0, ..., r-1\} \), where \( r \) is determined by the encrypter. It will encrypt the message M and gives out a ciphertext CT and verification set VR. So that only when the user’s attributes satisfy the corresponding privilege tree \( Tp \) he can execute specific operation on the cipher text.

Encrypt (PK, M, \{Tp\} \( p \in \{0, ..., r-1\} \) ) → (CT, VR).

4) Decrypt: This algorithm takes the public key PK, a ciphertext CT, and a private key SKu, which has a set of attributes Au and corresponds to its holder’s GIvMs as input. If the set Au satisfies any tree in the set \{Tp\} \( p \in \{0, ..., r-1\} \), the algorithm gives out a message M or a verification parameter. If the verification parameter is successfully verified by Cloud servers, who use VR to verify it, the operation request will be processed.

Decrypt (PK, SKu, CT) → M or verification parameter Comparisons of different ABE algorithms are shown in Table 1.

<table>
<thead>
<tr>
<th>Scheme</th>
<th>Distributed</th>
<th>CP-ABE</th>
<th>KP-ABE</th>
<th>Decryption Out</th>
<th>Key Blind</th>
</tr>
</thead>
<tbody>
<tr>
<td>Junzuo Lai</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Aijun Ge</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Jin Li</td>
<td>X</td>
<td>X</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Jinguang Han</td>
<td>X</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Suhair Alshehri</td>
<td>X</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Changji Wang</td>
<td>X</td>
<td>✓</td>
<td></td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Shengyan Wu</td>
<td>✓</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Junzuo Lai</td>
<td>X</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td>X</td>
</tr>
</tbody>
</table>

TABLE 1: Comparisons of different ABE algorithms

III. CONCLUSION
The most important aspect that is to be considered in storing data is the security mechanisms associated with it. This paper describes about the highly efficient Attribute Based Encryption technique which provides security for appropriate users by using the user based access control attributes. This paper surveys different Attribute Based Encryption techniques available for performing flexible encryption on the data and explains the different algorithms followed in each of these techniques. Bilinear Maps, the essential criteria needed for understanding the basic Attribute Based Encryption is also focused. The different share outsourced data methods on untrusted storage server instead of trusted server are furnished. Finally, a comparison table on different ABE’s with respect to certain parameters has also been provided.

REFERENCES
[8] Dongxiao Liu, Hongwei Li Achieving, “Multi-Authority Access Control with Efficient Attribute Revocation in Smart Grid”, IEEE Communication...


