A Survey on ABE Based Secure Data Retrieval Schemes for DTN Networks

Fathima Nizar, G.S Santhosh Kumar
Department of Computer Science and Engineering
Amal Jyothi College of Engineering
Kottayam, India

Abstract— Disruption Tolerant Networks (DTN) technologies are becoming successful solutions that allow wireless devices carried by soldiers to communicate with each other and access the confidential information reliably. Some of the most challenging issues in this scenario are the enforcement of authorization policies and the policies update for secure data retrieval. This paper introduces a survey on ABE based secure data retrieval schemes for DTN networks. The different ABE schemes are discussed in this paper.

Keywords— Attribute Based Encryption (ABE), Cipher text Policy-Attribute Based Encryption (CP-ABE), Key Policy-Attribute Based Encryption (KP-ABE), Distributed Attribute Based Encryption (DABE), Disruption Tolerant Networks (DTNs).

I. INTRODUCTION

As the networks are developing widely, communication security over the Internet is becoming more important. Cryptography is one of the main field of research which is used to enhance the communication security. The various cryptography methods are DES, RSA, and ABE, which are exclusively used to encrypt, which is the process of converting plaintext into cipher-text. After data encryption, the secret data appears to be meaningless bits. Encryption avoids unauthorized user to decrypt or destroy it.

The Attribute Based Encryption (ABE) [5] is an approach that provides secure data retrieval in Disruption Tolerant Networks. This mechanism enables an access control over encrypted data using access policies and attributes among private keys and cipher texts. The Cipher text-Policy Attribute Based Encryption (CP-ABE) [2], which is one of the important type of ABE schemes, provides a scalable way of encrypting data such that the encryptor defines attribute set that the decryptor needs to possess in order to decrypt the cipher text.

However, the problem of applying the ABE to DTNs introduces several privacy and security challenges. The first challenge is the key revocation problem. Some users may change their attributes at some point of time, so key revocation for each attribute is necessary in order to make the systems secure. But, this problem is more difficult in ABE systems, since multiple users shares each attribute. Therefore revocation of any attribute or any single user in an attribute group may affect the other users in the group.

The key escrow problem is another challenge. In CP-ABE, the private keys of users are generated by the key authority, by generating their attribute keys. This could be a potential threat to the privacy or data confidentiality, if the key authority is compromised by some adversaries.

Disruption Tolerant Networking [6] is a networking architecture that is designed to provide communications in most unstable and stressed environments. This networks are frequently used in peace-keeping missions, disaster relief missions, and in vehicular networks. Most recently NASA has tested DTN technology for spacecraft communications [6]. The different features [6] of DTNs are:

- Fault-tolerant methods and technologies
- Electronic attack recovery
- Degradation quality from heavy traffic loads
- Minimal latency due to unreliable routers

Fig. 1 shows the architecture of the DTN. The architecture consists of the following system entities [1].

---

ISSN: 2231-5381  http://www.ijettjournal.org  Page 39
**Key Authorities:** They are the key generation centers that generate public or secret parameters for CP-ABE. The key authorities consist of central authority and multiple local authorities. There are secure and reliable communication channels between a central authority and each local authority. Each local authority manages different attributes and issues corresponding attribute keys to users.

**Storage node:** This entity stores data from senders and provide corresponding access to users. It may be mobile or static.

**Sender:** This entity owns confidential data and wishes to store them into the external data storage node. A sender is responsible for defining access policy and encrypts the data under the policy before storing it to the storage node.

**User:** This is a mobile node that wants to access the data stored at the storage node. If a user possesses a set of attributes satisfying the access policy of the encrypted data then he will be able to decrypt the ciphertext and obtain the data.

## II. ATTRIBUTE BASED ENCRYPTION SCHEMES

An ABE cryptosystem is designed to enable fine-grained access control of the encrypted data. It allows the encryptor to attach attributes or policies to a message being encrypted so that only the receiver who is assigned compatible policies or attributes can decrypt it. Formally, the attributes can be considered as boolean variables with arbitrary labels, and the policies are expressed as conjunctions and disjunctions of attribute variables. The ABE systems can be viewed as a generalization of Identity Based Encryption (IBE) systems [7]. In IBE systems, only one attribute is used which is the identity of the receiver, whereas ABE systems enable the use of multiple attributes simultaneously [2].

We have two alternatives in enforcing the access policy. The access policy can be embedded in the private key of a user, which results in a cryptosystem called Key Policy Attribute-Based Encryption (KP-ABE). Alternatively, the access policy can be embedded in the cipher text, which results in the Cipher text Policy Attribute Based Encryption (CP-ABE) system. Both KP-ABE and CP-ABE schemes ensure that a group of users cannot access any unauthorized data by colluding with each other [2].

### A. Key Policy Attribute-Based Encryption

In a key-policy attribute-based encryption (KP-ABE) system, cipher texts are labeled by sender with a set of descriptive attributes, while user's private key is issued by trusted attribute authority captures a policy that specifies which type of cipher texts the key can decrypt. KP-ABE schemes are suitable for structured organizations with rules about who may read particular documents. Typical applications of KP-ABE are secure forensic analysis and target broadcast [8].

A KP-ABE scheme is parameterized by a universe of possible attributes U and consists of the following four algorithms [8].

**Setup:** this algorithm is run by the trusted attribute authority, which takes as input the security parameter \( \lambda \) and the attribute universe U. It outputs some public parameters \( params \) and the master secret key \( msk \). The trusted attribute authority publishes \( params \) and keeps \( msk \) secret.

**KeyGen:** this algorithm run by the trusted attribute authority, which takes as input the public parameters \( params \), master secret key \( msk \), and an access structure \( A \) which is assigned by the trusted attribute authority to the user. It outputs a decryption key \( SK_A \).

**Encrypt:** this algorithm run by the sender, which takes as input the public parameters \( params \), a set of descriptive attributes \( W \), and a message \( m \in \{0,1\}^* \). It outputs the ciphertext \( c \).

**Decrypt:** this algorithm run by the recipient, which takes as input the public parameters \( params \), the ciphertext \( c \) that was encrypted under the set of attributes \( W \), and the decryption key \( SK_A \) for access structure \( A \). It outputs the message \( m \) if \( W \subseteq A \).

### B. Ciphertext Policy Attribute-Based Encryption

In a ciphertext-policy attribute-based encryption (CP-ABE) system, when a sender encrypts a message, they specify a specific access policy in terms of access structure over attributes in the ciphertext, stating what kind of receivers will be able to decrypt the ciphertext. Users possess sets of attributes and obtain corresponding secret attribute keys from the attribute authority. Such a user can decrypt a ciphertext if his attributes satisfy the access policy associated with the ciphertext. Thus, CP-ABE mechanism is conceptually closer to traditional role-based access control method [8].

**Access Structure**

Each leaf node of the access tree \( T \) represents either a positive or a negative attribute. Each internal node of \( T \) represents a threshold gate, which can be an ‘AND’ gate or an ‘OR’ gate in special cases. If \( num_x \) is the number of children of a node \( x \) and \( k_0 \) is its threshold value, then \( 0 < k_0 \leq num_x \). We note that a threshold gate represents an OR gate if \( k_0 = 1 \), and an AND gate if \( k_0 = num_x \). We denote the parent of node \( x \) by \( parent(x) \). The function \( att(x) \), which is defined only for the leaf nodes, denotes the attribute associated with a leaf node \( x \). In the access tree, we also define an ordering among the child nodes of every node. The children of node \( x \) are numbered from 1 to \( num_x \). We denote such a number associated with node \( x \) by \( index(x) \) [2].
We denote by $T_x$ the subtree of $T$ rooted at node $x$. If a set of attributes $S$ satisfies the access tree $T_x$, we denote it as $T_x(S) = 1$. We compute $T_x(S)$ recursively as follows. If $x$ is a non-leaf node, evaluate $T_x(S)$ for each child node $c_i$ of $x$. $T_x(S)$ returns 1 if and only if at least $k_0$ child nodes return 1, where $k_0$ is the threshold value of node $x$. If $x$ is a leaf node and denotes a positive attribute, then $T_x(S)$ returns 1 if and only if $\text{att}(x) \in S$. On the other hand, if $x$ is a leaf node and denotes a negative attribute, then $T_x(S)$ returns 1 if and only if $\text{att}(x) \notin S$ [2].

In the CP-ABE [2] scheme, each user is associated with a set of attributes and his private key is generated based on these attributes. When encrypting a message $M$, the encryptor specifies an access structure which is expressed in terms of a set of selected attributes for $M$. The message is then encrypted based on the access structure such that only those whose attributes satisfy this access structure can decrypt the message. Unauthorized users are not able to decrypt the ciphertext even if they collude. In [6], the access structure is sent in plaintext. A CP-ABE scheme consists of the following four algorithms [1] [2]:

**System Setup**

This is an algorithm that takes a security parameter as input, and outputs public parameters $PK$ and a master key $MK$. $PK$ is used for encryption and $MK$ is used to generate user secret keys and it is known only to the central authority.

At initial system setup phase, the trusted initializer chooses a bilinear group $G_0$ of prime order $p$ with generator $g$ according to the security parameter. It also chooses hash functions $H : \{0, 1\}^* \rightarrow G_0$ from a family of universal one-way hash functions.

The public parameter $\text{param} \text{ is given by } (G_0, g, H)$. For brevity, the public parameter $\text{param}$ is omitted below.

1) **Central Key Authority:** CA chooses a random exponent $\beta \in \mathbb{Z}_p^*$. It sets $h = g^\beta$. The master public/private key pair is given by ($PK_{\text{CA}} = h$, $MK_{\text{CA}} = \beta$).

2) **Local Key Authorities:** Each $A_i$ chooses a random exponent $\alpha_i \in \mathbb{Z}_p^*$. The master public/private key pair is given by ($PK_{A_i} = e^{(g, g)^{\alpha_i}}$, $MK_{A_i} = \alpha_i$).

**Encryption**

This is an algorithm that takes as input a message $M$, an access structure $T$, and the public parameters $PK$. It outputs the ciphertext $CT$.

Let $Y$ be the set of leaf nodes in the access tree. To encrypt a message $M \in G_1$ under the access tree $T$, it constructs a ciphertext using public keys of each authority as follows.

\[ \text{CT} = (T, \hat{C} = \text{Me}(g, g)(\alpha_1 + \ldots + \alpha_m)^y, Y), \forall y \in Y : C_y = g^{t(y)}, C_y' = H(\lambda_y) \times (g^{t(y)})^{\beta}, \]

Where $\hat{C}$ can be computed as $\hat{C} = \text{Me}(PK_{A_1} \star \ldots \star PK_{A_m})^y = \text{Me}(g, g)(\alpha_1 + \ldots + \alpha_m)^y$.

**Key Generation**

This is an algorithm that takes as input the set of a user (say $X$)’s attributes $S_X$, the master key $MK$ and outputs a secret key $SK$ that identifies with $S_X$.

In CP-ABE, user secret key components consist of a single personalized key and multiple attribute keys. The personalized key is uniquely determined for each user to prevent collusion attack among users with different attributes. The proposed key generation protocol is composed of personal key generation followed by attribute key generation protocols. It exploits arithmetic secure 2PC protocol to eliminate the key escrow problem such that none of the authorities can determine the whole key components of users individually.

**Personal Key Generation:** The central authority and each local authority are involved in the following protocol.

1) When CA authenticates a user $u$, it selects random exponents $\gamma_1 + \ldots + \gamma_m \in \mathbb{Z}_p^*$ for each authority $A_1, \ldots, A_m \in A$; and $\tau_i = \sum_{i \in \lambda} r_i$. This $r_i$ value is a personalized and unique secret to the user. Then, CA and each $A_i$ engage in a secure 2PC protocol, where CA’s private input is $(\gamma_1, \beta)$, and $A'_i$’s private input is $\alpha_i$. The secure 2PC protocol returns a private output $x = (\alpha_i + \gamma_i) \beta$ to $A_i$. This can be done via a general secure 2PC protocol for a simple arithmetic computation.

2) $A_i$ randomly picks $I_i \in \mathbb{Z}_p^*$. Then, it computes $T = g^{x(I_i)} = g^{((\alpha_i + \gamma_i) \beta) + I_i}$ and sends it to CA.

3) CA then computes $B = T^{1/\beta} = g^{\alpha_i + \gamma_i \beta}$ and sends it to $A_i$.

4) $A_i$ outputs a personalized key component $D_i = B^\beta = g^{\alpha_i + \gamma_i \beta}$ and sends it to the user $u_i$ securely.

Then, the user $u_i$ computes its personal key component $D = \prod_{i=1}^{m} D_i = g^{((1 + m) + r) \beta}$.

**Attribute Key Generation:** After setting up the personalized key component, each $A_i$ generates attribute keys for a user $u_i$ with public parameter received from CA as follows.

1) $A_i$ first selects a random $r_i$, and sends $g^{\gamma_i \beta}$ and $g^{\tau_i}$ to $A_i$ and $u_i$, respectively.

2) $A_i$ takes a set of attributes $A_i \subseteq A_i ($Z$)$ as inputs and outputs a set of attribute keys for the user that identifies with that set $A_i$. It chooses random $I_i \in \mathbb{Z}_p^*$ for each attribute $\lambda_i \in A_i$. Then, it gives the following secret value to the user $u_i$: $\forall \lambda_i \in A_i : D_i = g^{x(I_i)}. H(\lambda_i)^{D_i(i)}, D_j = g^{\gamma_j}$.

Then, the user computes $g^{x(I_i)}$. $D_i$ for all its attributes key components and finally obtains its whole secret key set as $SK_u = (D = g^{x(I_i)} + \ldots + \beta), \forall \lambda_i \in S : D_i = g^{x(I_i)}. H(\lambda_i)^{D_i(i)}, D_j = g^{\gamma_j}$.

Where, $S = \bigcup_{i=1}^{m} A_i$. 

ISSN: 2231-5381 http://www.ijettjournal.org Page 41
Decryption
This algorithm takes as input the ciphertext CT, a secret key SK for an attribute set SX. If SX satisfies the access structure embedded in CT, it will return the original message M.

When a user receives the ciphertext CT from the storage node, the user decrypts the ciphertext with its secret key. We first define a recursive algorithm DecryptNode(CT,SK,x) that takes as inputs a ciphertext CT, a private key SK, which is associated with a set A of attributes, and a node x from the tree T. It outputs a group element of G or ⊥.

We suppose that a user ut performs the decryption algorithm.

If x is a leaf node, then define as follows. If λt ∈ A, then

\[
\text{DecryptNode}(CT, SK, x) = e(Dx, Cx) = e((g^{\tau x} \cdot H(\lambda_t)^{\tau x}, g^{\tau x(0)}))
\]

If λt /∈ A, we define DecryptNode(CT, SK, x) = ⊥.

C. Distributed Attribute-Based Encryption

The system is built up of a Central Authority (CA) and multiple Attribute Authorities (AAs). These attribute authorities separately maintain attributes. The major components of the scheme are master, users and attribute authorities. The duty of the master is to distribute private user keys. Attribute Authority certifies the user and distributes private attribute key to the user that can be used for decrypting ciphertext. User produces ciphertext by the method of encryption. Whenever needed user decrypts the ciphertext and retrieves the original message [9].

The following algorithms [9] are defined in a DABE scheme.

Setup: This algorithm generates the public key PK and the master key MK.

CreateUser: The outputs of this algorithm are a public user key PKu and a secret user key SKu.

CreateAuthority: This algorithm generates a private authority key SKa.

RequestAttribute PK: This algorithm generates the public attribute key of attribute A.

RequestAttribute SK: This algorithm generates a secret attribute key SK_{A, u} for user u.

Encrypt: The inputs of this algorithm are public key, message, an access policy and the public keys associated with the attributes in the access policy. The output of this algorithm is the ciphertext.

Decrypt: The inputs of the algorithm are the ciphertext produced by the Encrypt algorithm, an access policy and a key ring. Decryption is performed based on certain conditions and if the conditions are satisfied, the algorithm will output the plaintext.

The advantage [9] of the DABE scheme is that only two pairing operations are required in the decryption algorithm. Pairing operations are the most expensive operations in cryptography. The efficiency of the scheme can be improved by reducing the number of pairing operations. Only the decryption algorithm requires pairing operations. The pairing operations are not used anywhere else.

The major disadvantage of DABE scheme is that the overhead involved in managing the distributed authorities.

D. Decentralized Attribute-Based Encryption

The decentralized attribute-based encryption was proposed by Lewko and Waters [3]. This system [9] is a multi-authority attribute based encryption system.

Features
1. Any party can be appointed as an authority.
2. Global coordination of the authorities is required only for creating an initial set of common reference parameters.
3. To become an ABE authority, the party creates a public key and distributes private keys to users.
4. The encryption of the data is performed by the user with the help of a Boolean formula.
5. No central authority is required in the system.

Scalability, Security and Efficiency

The overhead of relying on a central authority is eliminated from the system. This ensures more scalability for the system. Without central authority, the security and efficiency of the system much more compares to other systems.

Robustness

Since the authorities are working independently of each other, the failure of one authority will not affect the working of other authorities. This improves the robustness of the system.

Autonomous key generation and Collusion resistance

The system uses a hash function on user’s global identifier so that collusion resistance is ensured for multiple keys generated by different authorities.
The system uses the following algorithms [9]:

**Global Setup:**
This algorithm chooses a bilinear group G of order N. This N acts as a component of the global public parameter. The output of the algorithm is a description of a hash function.

**Authority Setup:** Each authority chooses two random exponents for the attribute i. The output of this algorithm is a public key and a secret key.

**Encrypt:** The inputs of this algorithm are a message, an access matrix, the global parameters and the public keys of the authorities. The output of this algorithm is the ciphertext.

**KeyGen:** A key is created for the (identity, attribute) pair.

**Decrypt:** A hash function is applied on the identity for attribute i. The output of this algorithm is the message.

The advantage of the system is that the system provides collusion resistance and the system is more efficient, more robust and provides scalability.

### III. CONCLUSION

This paper introduced a survey on ABE based secure data retrieval schemes for DTN networks. The different ABE schemes such as Attribute Based Encryption (ABE), Ciphertext Policy-Attribute Based Encryption (CP-ABE), Key Policy-Attribute Based Encryption (KP-ABE), Distributed Attribute Based Encryption (DABE), Decentralized Attribute Based Encryption are discussed. This schemes can be used in different ways to provide security for data retrieval in DTNs.

Ciphertext Policy-Attribute Based Encryption is the better one among this schemes.

### REFERENCES


