ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 1, January 2017

A Literature Survey on Attribute-based Encryption Schemes of Access Policy in Shared Environment

Mr. S. B. Vhotkar[1], Prof. R. V. Argiddi[2]

P.G. Student, Dept. of CSE, Walchand Institute of Technology, Solapur, India[1]

Dept. of CSE, Walchand Institute of Technology, Solapur, India [2]

ABSTRACT: In Attribute-based Encryption (ABE), attributes are having importance as they determine public key for encrypting data and can be used as an access policy to control users' access. There are two types of access policy; key-policy and ciphertext-policy. The key-policy is the access structure on the user's private key, and the ciphertext-policy is the access structure on the ciphertext. The ABE scheme has two advantages (1) it reduces the communication overhead of the Internet, and (2) it provides a fine-grained access control. In this paper comparisons of various schemes are carried out on certain criteria. The schemes covered here are a basic attribute-based encryption, two various access policy attribute-based encryption and two various access structures.

KEYWORDS: Cloud computing, attribute-based encryption, access control, fine-grained access, revocation.

I. Introduction

Cloud computing technology is growing with rapid pace and lots of people are uploading their data on cloud to process, store, or to share it with other users as the cloud is providing various application services to satisfy users' requirement [1]. For providing the application services to the end user, cloud environment is charging nominal charges to them. Due to its hassle free services and cost efficiency, more and more people are getting attracted towards cloud computing environment.

With advantages some disadvantages follows; the data owner has to make a flexible and scalable access control policy so that only authorized users can access. Cloud environment is inherently insecure in nature. To protect the data, it must be encrypted before uploading it on Cloud Storage. Traditional public key infrastructure can be used but it has problems: (1) to be able to encrypt data, the data owner needs to obtain the data user's public key to complete this; (2) a lot of storage overhead because of the same plaintext with different public keys.

To rectify these problems, Sahai and Waters proposed an attribute-based encryption (ABE) scheme in 2005, and this is the first attempt of proposing the concept of the attribute-based encryption scheme [5]. In this scheme, user's identity is used as attributes, and a set of attributes are used to encrypt and decrypt data.

In 2006, Goyal et al. proposed a Key-Policy Attribute Based Encryption (KP-ABE) scheme that built the access policy into the user's private key and described the encrypted data with user's attributes [3]. The KP-ABE scheme provides fine-grained access control and more flexibility to control users than ABE scheme. But the disadvantage of KP-ABE is that the access policy is built into user's private key, so data owner can't choose who can decrypt the data except choosing a set of attributes which can describe this data. It is unsuitable in certain application because data owner needs to trust the key issuer. Due to its inability to express negative attributes, it can't exclude the unwanted users from accessing the data. So, Ostrovsky et al. proposed a non-monotonic access structure in 2007, and this scheme allow each attribute attach a tag in front of them [6]. And Bethencourt et al. also proposed a ciphertext-policy attribute based (CP-ABE) scheme in the same year, and the CP-ABE scheme built the access policy into the encrypted data; a set of attributes is in user's key. The CP-ABE scheme addresses the problem of KP-ABE that data owner only trusts the key issuer. After that, several schemes were proposed based on the CP-ABE scheme.

The Criteria of an ideal Attribute-based encryption Scheme

DOI: 10.15680/IJIRCCE.2017. 0501124

C1. Data Confidentiality:

Before uploading data to the cloud, the data was encrypted by the data owner. Therefore, unauthorized parties including the cloud cannot know the information about the encrypted data.

C2. Fine grained access control:

In the same group, the system granted the different access right to individual user. Users are on the same group, but each user can be granted the different access right to access data. Even for users in the same group, their access rights are not the same.

C3. Scalability:

When the authorized users increase, the system can work efficiently. So the number of authorized users cannot affect the performance of the system.

C4. User accountability:

If the authorized user is dishonest, he would share his attribute private key with the other unauthorized user. It causes the problem that the illegal key would share among unauthorized users.

C5. User revocation:

If the authorized user is dishonest, he would share his attribute private key with the other unauthorized user. It causes the problem that the illegal key would share among unauthorized users.

C6. Collusion resistant:

Users cannot combine their attributes to decipher the encrypted data. Since each attribute is related to the polynomial or the random number, different users can not collude each other.

II. Related work

A) ATTRIBUTE-BASED ENCRYPTION SCHEME

Sahai and Waters proposed an attribute based encryption scheme in 2005. In this scheme, there are authority, data owner (also be called sender) and data user (also be called receiver) and authority generate keys for data owners and users to encrypt or decrypt data. The authority generates keys according to attributes; and these attributes of public key and master key, which are generated by the authority, should predefine (means that it will list attributes which will be used in the future). If any data user who wants to be part of this system, and he owns attributes that are not part of the predefined attribute, the authority will re-define attributes and generate a public key and master key again. Data owner's role in this scheme is to encrypt data with a public key and a set of descriptive attributes. Data user decrypt encrypted data with his private key sent from the authority, and then he can obtain the needed data. For decrypting data, attributes in data user's private key will be matched with the attributes in encrypted data. If the number of "matching" is at least a threshold value, the data user's private key will be permitted to decrypt the encrypted data. For example, for a set of descriptive attributes in the encrypted data, {MIS, Teacher, Student}, the threshold value is 2. If a data user wants to decrypt the encrypted data, the number of attributes in his private key should have two or more than two of attributes in the encrypted data. So, a data user has a private key with attributes, {MIS, Student} to decrypt and obtain the data for the above mentioned access policy.

This scheme has four algorithms: Setup, KeyGen, Encrypt, and Decrypt. Let G1 and G2 be two bilinear groups of prime order p, and let g be a generator of G1. In addition, let $e : G_1 \times G_1 \rightarrow G_2$denote the bilinear map, and let d be a threshold value.

1)Setup(d): The authority uniformly and randomly chooses $t_1, \ldots, t_n$, y from Zq, and publishes the public key, PK = ($T_1=g^t{}_1, \ldots, T_n=g^t{}_n$, Y = $e(g,g)^y$), And the master key is MK=($t_1, \ldots, t_n$,y).

2) KeyGen(A_U, PK, MK): The authority executes and generates a private key for the data user U. Choose a (d – 1) degree polynomial q randomly such that q(0) = y. The data user’s private key D is $\left\{Di = g^{\frac{q(i)}{ti}}\right\}_{\forall i \in A_U}$.

3)Encrypt(A_{CT}, PK, M): Data owner encrypts message $M \in G_2$ with a set of attributes A_{CT}. Choose a random number $s \in Z_q$, and the encrypted data is published as CT = (A_{CT}, E = MYs= $e(g,g)^{ys}$, $\{E_i = g^{t_i s}\}_{\forall i \in A_U}$).

4)Decrypt(CT, PK, D): Data user decrypts the encrypted data CT with the private key D. Choose d attributes from $i \in A_{CT} \cap A_{CT}$ to compute e(Ei,Di)=$e(g,g)^{q(i)s}$ if$|A_{CT} \cap A_{CT}| \geq d$. And compute Ys = $e(g,g)^{q(0)s}$ = $e(g,g)^{ys}$ with the Lagrange coefficient and the message $M = \frac{E}{Y^s}$ can be obtained.

In KenGen() algorithm, the user's private key is generated with secret sharing. The shares of secret y are embedded in the components of the user's private key Di, and the secret key is associated with the random polynomial q(i). So every user's private key D cannot be combined to a new private key to perform the collusion attack. And in the Encrypt() algorithm, the random number s can avoid user decrypting the data after the first decrypting, when he infers the number. Besides, the component of the encrypted data Ei would be used in Decrypt() algorithm, the needed attributes can be known through this component. The attributes in user's private key and the encrypted data can let this scheme achieve access control. The authorized users can use their private key to decrypt the corresponding data. In addition, application of this scheme would be restricted in the real environment because it uses the access of monotonic attributes to control user's access.

B) KEY-POLICY ATTRIBUTE-BASED ENCRYPTION SCHEME

In 2006, Goyal proposed a key-policy attribute-based (KP-ABE) scheme. In this scheme, set of descriptive attributes are assigned to the encrypted data and access policy is built in user's private key. If attributes of the encrypted data can satisfy the access structure in user's private key D, user can obtain the message using decryption algorithm. In addition, the KeyGen() algorithm is different from the attribute-based encryption which is introduced at subsection one in this section. The user's private key to be generated depends on the access structure. This algorithm uses secret sharing and chooses a polynomial q_x such that $q_x(0) = q_{parent(x)}(index(x))$, (Here parent(x) is x's parent node, and index(x) is the number associated with node x that is given by x's parent node.) in a top-down manner which is to start from the root node r for each node x in the access structure. So qr(0) is equal to the master key y, and the master key y is distributed among the user's private key component Di which is corresponding to the leaf node (Note that the leaf node represents attribute).

This scheme has four algorithm for encryption and decryption process: Setup, KeyGen, Encrypt, and Decrypt.

1)Setup(d): The authority selects several uniform and random numbers from Zq and generates public key PK and master key MK.

2)KeyGen(A_{U-KP},PK,MK): Authority generates private key components for each leaf node in the access structure. These components will be merged into the user's private key, and be sent to user.

3)Encrypt(M,A_{CT},PK):Data owner selects a random number and encrypts s message M with a set of attributes and then he generates the encrypted data as CT.

4)Decrypt(CT,D):This algorithm executed recursively to obtain the user data from encrypted data. For that it takes encrypted data, user's private key, and nodes of the access structure in user's private key.

C) CIPHERTEXT-POLICY ATTRIBUTE-BASED ENCRYPTION SCHEME

In 2007, Bethencourt et al. proposed a ciphertext policy attribute-based scheme. In CP-ABE, access policy is built over the descriptive attribute and is attached to the ciphertext [2]. Descriptive attributes are attached to user's private key. If the set of descriptive attributes associated with the private key satisfies the access policy of the ciphertext, the user's private key can decrypt the underlying ciphertext. Just like KP-ABE, this scheme also has algorithms, Setup(), KeyGen(), Encrypt(), Decrypt() and Delegate().

The parameters in this scheme and KP-ABE are the same. The scheme will be described as follows.

1)Setup : The authority chooses two random number alpha, beta from Zq as exponents, and generates the public key, PK = (G_0, g, h=g^{β} , f =$g^{1/\beta}$, $e(g,g)^{\alpha}$). The master key is MK=(β, g^{α}).

2)KeyGen(MK, A_U): The authority chooses a random number s from Zq and random sj for each attribute j in a set of attributes in user's private key. The user's private key, $D = (DK = g^{\frac{(\alpha+s)}{\beta}}, \forall j \in A_U : D_j = g^s \cdot H(j)^{s_j}, D_j^* = g^{s_j})$ is output.

3)Encrypt(PK, M, A_{CT-CP}) : Data owner uses this algorithm to encrypt the message M with the access structure A_{CT-CP}. Choose a random number y∈Z_q, set $q_r(0)$=y, where r is the root node, and let I be the set of leaf nodes in A_{CT-CP}. The message is encrypted with access structure A_{CT-CP}, and then outputs the encrypted data, $\mathrm{CT} = \left(A_{CT-CP}, \tilde{C} = Me(g,g)^{\alpha y}\right), C = h^y, \forall i: C_i = g^{q_i(0)}, C_i^* = H(att(i))^{q_i(0)}$.

4)Delegate(D, $\widetilde{A_U}$): This algorithm takes the user's private key D and a set of attributes whose each attribute is in A_U to create a new user's private key $\widetilde{D}$.

5)Decrypt(CT, D): When data user receives the encrypted data, he can execute this algorithm. The user's private key D and the encrypted data are input in this algorithm and the recursive function, and decryptnode is called. If the node x is leaf node and let k=att(x), where k ∈ A_U, the decryptnode can be called, then it computes decryptnode(CT, D, x) = $\frac{e(D_k, C_x)}{e(D_k^*, C_x^*)}$) = $e(g,g)^{sq_x(0)}$. If k is not in A_U, decryptnode will output invalid. If x is not the leaf node, the decryptnode function can be called and all children nodes of node x, z can be input to execute. It use Lagrange coefficient to compute and obtain $e(g,g)^{sq_x(0)}$. Hence, if the access structure A_{CT-CP} satisfies A_U, the CT, D, s are input to compute decryptnode(CT, D, s)=$e(g,g)^{sy}$. The algorithm can decrypt by computing $\frac{\tilde{C}}{e(C,DK)/e(g,g)^{ys}} = M$ to recover the message M.

The CP-ABE builds the access structure in encrypted data thus allowing it to choose who can decrypt. It supports the access control in the real environment.

D) ATTRIBUTE-BASED ENCRYPTION SCHEME WITH NON-MONOTONIC ACCESS STRUCTURES

In 2007, Ostrovsky et al. proposed an attribute-based encryption with non-monotonic access structure [6]. The scheme we studied so far has no negative Boolean operator. This scheme introduces a new Boolean formula operator, NOT to make it non-monotonic access control structure. Thus this scheme defines access structure {MIS AND Student AND NOT graduate} which allows to decrypt the encrypted data through private key that has MIS and student but not graduate as its attributes.

This scheme contains four algorithms: Setup(), KeyGen(), Encrypt(), and Decrypt().

1)Setup(d): The parameter d is number of attributes used for encrypting the data. Let G_1 be a bilinear group of prime order p, let g be a generator of G_1, and let e:G_1xG_1→G_2 denote the bilinear map. Choose two random numbers α, β from Zq, and denote $g_1=g^\alpha$, $g_2=g^\beta$. Let h(x), q(x) be two polynomials of degree d and constraint that q(0)= β. Generate the public key, PK=(g, g_1; $g_2=g^{q(0)}$, $g^{q(1)}$, $g^{q(2)}$,…, $g^{q(d)}$; $g^{h(0)}$, $g^{h(1)}$, $g^{h(2)}$,…, $g^{h(d)}$), and the master key, MK= α. In addition define two publicly computable function, T,V:Zq→G_1 such as T(x)→$g_2{}^{xd}.g^{h(x)}$, V(x)→$g^{q(x)}$.

2)KeyGen(~A_U, PK, MK): The authority use this algorithm to generate key for various users. ~A_U is non-monotonic access structure, select a random number, si ∈ Zq, for each attribute xi. P(x) is randomly selected polynomial such that P(0)= α. It outputs user's private key.

3)Encrypt(M, A_{CT}, PK): Data owner encrypt his message M ∈ G_2 under a set of attributes A_{CT} ⊂ Zq. A random number s is chosen from Zq to compute the encrypted data CT.

4)Decrypt(CT,D): It takes encrypted data CT and private key D of a user as input. First, the data user checks if A_{CT} ∈ A_U. If not, its output is invalid or else original message M is outputted.

E) HIERARCHICAL ATTRIBUTE-BASED ENCRYPTION SCHEME

In 2011, Wang et al. proposed a hierarchical attribute-based encryption scheme. Keys are generated with the key generation algorithm of HIBE [4]. It used disjunctive normal form(DNF) to express the access control policy. There are five roles in this scheme: the cloud storage service, data owner, the root authority, the domain authority, and data users. Cloud storage service takes care of the storing and sharing of user's data with others. The Data owner is responsible for encrypting data and sharing it with others. The root authority generates system parameters and domain keys to distribute them. The domain authority manages the domain authority at net level and all users in its domain, to delegate keys for them. Besides, it can distribute secrete keys for users. User use their secrete key to decrypt the encrypted data and obtain the message. The key generation is done hierarchically. At the first level, root authority generates a root master key for domain authority. The system public key and the master key of the domain authority at first level are used to create the master keys for the domain authorities at the next level by the root authority or the domain authority at the first level. The algorithms of this scheme are as follows.

1)Setup(K): The security parameter are chosen. Using these parameters, this algorithm generates system public key(PK) and the system master key(MK). The system public key(PK) is kept open for all the authorities as well as users, but the system master key is kept secret.

2)CreateDM(PK, MKi, PKi+1): The root authority or the domain authority generates master keys MKi+1 for domain authoritiesDMi+1 by using system public key(PK), the public key of domain authoritiesDMi+1, PKi+1 and its master key MKi.

3)CreateUser(PK,MKi,PKu,PKa): The domain authority first ensures whether the user u is authorized for attribute a which is monitored by itself. If so, it creates the user identity secret key Di,u and the user attribute secret key Di,u,a.

4)Encrypt(PK,M,Act):The data owner encrypts data M with a DNF access control policy and public keys of all attributes in access control policy and outputs CT as encrypted data.

5)Decrypt(PK,CT,Di,u):This algorithm checks whether the users attributes satisfies the access policy in the encrypted data. If so, then user's message is generated by this algorithm.

III. COMPARISONS

In this section, we compared the above discussed scheme by the criteria: fine-grained access control, data confidentiality, scalability, user accountability, user revocation and collusion resistance. Table 1 shows us that HIBE encryption scheme satisfies all the criteria then the number comes of CP-ABE, KP-ABE, ABE with Non-monotonic and lastly ABE.

Table 1: The criteria of an ideal attribute-based encryption scheme

Criteria	ABE	KP-ABE	CP-ABE	ABE with Non-monotonic	HIBE
Fine-grained access control	N	Y	Y	Y	Y
Data confidentiality	Y	Y	Y	Y	Y
Scalability	N	N	N	N	Y
User accountability	N	N	Y	N	Y
User revocation	N	Y	Y	Y	Y
Collusion Resistance	Y	Y	Y	Y	Y

IV. CONCLUSION

After doing survey of different attribute-based encryption schemes like ABE, KP-ABE, CP-ABE, ABE with non-monotonic access structure, and HABE, we can say that these schemes can be classified according to their access policy. The comparative study reveals, as shown in the Table 1, that every attribute-based encryption schemes has their own pros and cons and their suitability depends on the private cloud environment under consideration for example ABE schemes (like CP-ABE and KP-ABE) are generally useful in the field of proxy re-encryption.

REFERENCES

1. M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "A view of cloud computing," Communications of the ACM, vol. 53, pp. 50-58, 2010.
2. J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in Proceedings of IEEE Symposium on Security and Privacy, pp. 321-334, 2007.

3. V. Goyal, O. Pandey, A. Sahai, 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, pp. 89-98, 2006.
4. G. Wang, Q. Liu, and J.Wu, "Hierarchical attribute-based encryption for fine-grained access control in cloud storage services," in Proceedings of the 17th ACM conference on Computer and communications security, pp. 735-737, 2010.
5. A. Sahai and B. Waters, "Fuzzy identity based encryption," Advances in Cryptology V EUROCRYPT, vol. 3494 of LNCS, pp. 457-473, 2005.
6. R. Ostrovsky, A. Sahai, and B. Waters, "Attribute-based encryption with non-monotonic access structures" in Proceedings of the 14th ACM conference on Computer and communications security, pp. 195-203, 2007.
7. S. Yu, C. Wang, K. Ren, and W. Lou, "Achieving secure, scalable, and fine-grained data access control in cloud computing," in Proceedings of IEEE INFO-COM, pp. 534-542, 2010.
8. S. Yu, C. Wang, K. Ren, and W. Lou, "Attribute based data sharing with attribute revocation," in Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security, pp. 261-270, 2010.
9. B. Waters, "Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization," Public Key Cryptography V PKC, vol. 6571 of LNCS, pp. 53-70, 2011.
10. Q Li, H. Xiong, F. Zhang, and S. Zeng, "An expressive decentralizing KP-ABE scheme with constant-size ciphertext," International Journal of Network Security, vol. 15, no. 3, pp. 161-170, 2013.
11. G.Wang, Q. Liu, J.Wu, and M. Guo, "Hierarchical attribute-based encryption and scalable user revocation for sharing data in cloud servers," Computer & Security, vol. 30, pp. 320-331, 2011.
12. A. Shamir, "How to share a secret," Communications of the ACM, vol. 22, pp. 612-613, 1979.