A scheme supporting efficient attribute revocation for cloud storage based on CPABE

Bodong Cai, LiaoTe Xiong, Junyan Ye, Meng Ma, Zhuo Tang
College of Computer Science and Electronics Engineering
Hunan University
Changsha, China
e-mail: turedss@gmail.com

***Abstract*—For low efficient of key management, most cloud centers lack fine-grained attribute revocation, which only support revocation of the keys. This paper proposes a novel access policy which supporting fine-grained attributes revocation based on CP-ABE. In this paper, we firstly re-design the key manage policy. The generation and distribubition of the user's private keys are completed by the collaboration of the data owner and attribute authority and it has reduced the cost of the key management. Secondly we embed a random parameter ED_x when publish the master key. Through the embeddoor, this scheme can control the access authority of users for who have lost the authority to acess the cloud data. Then, we can gurantee the security of keys based on the designed mechanism of binary tree. Our approach can achieve high performance and can also deal with fine-grained attribute revocation. Finally, the security of this policy has been proved under the standard model.**

Keywords- cloud storage; attribute-based encryption; binary tree; attribute revocation

I. INTRODUCTION

Cloud storage is derived and developed from the cloud computing as a storage technique. It has been widely used in storage areas because of low cost, simple interface and highly scalable. At the same time, security of cloud storage has been questioned while it's useful and convenient. According to a survey, nearly 70% of users are not willing to store their data in the cloud service provider [1]. More seriously, some well-known cloud service providers such as Google Docs and Linkup have been reported to suffer severe data leak scandals [2]. Obviously, how to enhance the security of the cloud storage has become a hot issue and a challenging in the community.

Access control is an effective method to achieve security and privacy of the user's data. As many cloud storage service provider provide only simple access control funcions, they can only guarantee the securiy of the user's data with tradtional encryption methods. While for the security of the data under the complex and changeable cloud enviroment, these providers often fail to achieve fine-grained acess control. To settle this problem, a new attribute-based encryption scheme is proposed by the reseachers. Sahai and Waters introduce a new method that utilizes an attribute-based encryption access control based on identity-based encryption [3, 4]. Then, Bethencourt and other researchers propose a new scheme called KPABE(key policy attribute-based encryption) based on ABE, but it cannot support NOT gate, and the attribute revocation is also not included [5]. The problem to design a flexible acess control scheme that can deal with both the NAND gate and the NOT gate has not been solved until Liang and Goyal propose a novel tree structure to eliminate the boundary condition [6, 7].

It is not a trival problem to design an excellent attribute revocation scheme for ABE. There are mainly three aspects need to be considered:

- ABE itself is a complex mechanism. Firstly, the ciphertext and the user keys are associated with a set of attributes. Secondly, although the access control policy about the ciphertext is decided by the data owners, the attribute keys are managed by the non-trusted attribute authorization.
- They are many-to-many relationships between the user groups and the attribute sets. The user groups are revoked along with the attribute revocation, vice versa. The attributes are revoked along with the revocation of the user groups.
- The revocation is complex. It includes user group revocation, user attribute revocation and system attribute revocation.

Pirretti is the first one to introduce the key revocation scheme for ABE [8], but the users have to keep close contact with the authorization center in order to obtain the key. Bethencourt et al [5] try to improve the performance of the key expire so as to reduce the traffic load of communication, but the shortage is that the attribute revocation cannot be done directly when it reaches the deadline. Then Yu et al introduce a semi-trusted proxy scheme based on proxy re-encryption technology, the application is limited as the service provider must be stay online [9].

In this paper a new CP-ABE scheme with embeddoor is proposed. Attribute authority is responsible for the generation and distribution of the user's secret key, while data owner is in charge of the generation and distribution of the data owner's secret key. The benefit of this approach is that it supports the access of data for both the users and the data owners. Besides, it can also reduce the computing cost and the cloud storage cost under the cloud environment. Meanwhile, we control user's access authority using the embeddoor by embedding a parameter to the MK. With the use of the embeddoor, we do not need to update the remained user's secret key during the attribute revocation, and it reduces the cost of re-encryption and achieves high performance for attribute revocation.

© 2014. The authors - Published by Atlantis Press

The remainder of our paper is structured as follows: In section 2 we introduce the pre-knowledge. In section 3 we define our system model and security model. In section 4 we define the binary tree mechanism and give our scheme. Then we analyze the security in section 5. Finally we give a conclusion in section 6.

II. PRE-KNOWLEDGE

A. Bilinear Maps

The bilinear maps are widely used in attribute-based encryption systems as it can make the cryptography scheme simple and efficient [10].

Definition 1: Let G_0 and G_1 be two multiplicative cyclic groups of prime order p, g be a generator of G_0 and e be a bilinear map, e: $G_0 \times G_0 \to G_1$ .The bilinear map e has the following properties:

- Bilinearity: for all $u,v \in G_0$ and $a,b \in Z_p$, we have $e(u^a, v^b) = e(u,v)^{ab}$.
- Non-degeneracy: $e(g,g) \neq 1$.
- Computability: for all $u,v \in G_0$, there exists an effective algorithm to calculat $e(u,v)$.

B. Ciphertext-Policy Attribute-Based Encryption

Definition 2: Attribute Sets, $A = \{1,2,3,\cdots k\}$ is the whole attribute sets, let attribute sets S be a nonvoid subset and $S \in A$, the subset of A considerd to be authorized and the subset don't belong to A considerd to be unauthorized.

Definition 3: linear secret sharing, let (M, ρ) represent the attributes policy p, where M is a matrix with the size defined as $\ell \times k$, ρ is a mapping function. $\rho(i)$ represents the attribute which associate with the M's ith row where $i = 1,2 \cdots \ell$ .Let S be a attribute set that satisfies policy p and let $\vec{M}$ be a vector which is composed by ith row of matrix M. We calculate a set of constant coefficients to satisfy the equation $\sum_{i \in I} \theta_i \vec{M}_i = \{1,0,\cdots,0\}$ ($\theta_i \in Z_p$) according to M. Whlie if S doesn't satisfy p, the constant coefficients cannot be calculated.

Let s be a secret sharing value. We randomly choose k-1 values from Z_p , they are denoted as $n_2, n_3, \cdots n_k$.Then we obtain a K dimensional vector $\vec{v} = (s, n_2, n_3 \cdots n_k)$ composed by s and n_2, $n_3, \cdots n_k$. Let $\lambda_i = \vec{M}_i \cdot \vec{v}$ be secret sharing values, then $\sum_{i \in I} \theta_i \lambda_i = s$. If the user's attibute policy of cloud satisfies the attribute policy p, then s can be recovered.

Definition 4: attribute groups

Let $U = \{u_1, u_2, u_3, \ldots\ldots, u_m\}$ be all the users' sets and $A = \{1,2,3,\ldots\ldots k\}$ be all the attribute sets, then we define attribute group $G(x)$ to represent the users who have the attribute x.

III. SYSTEM MODEL AND SECURITY MODEL

A. System Model

As shown in Fig.1, the system model presented in this paper consists of 4 parts: Data owner(DO), the trusted third party Attribute Authority（AA）, Cloud Storage Server (CSS) and users, referring to [12]. The system model is described as follows: DO is responsible for the generation and distribution of secret keys. AA is fully trusted and in charge of system initialization and generation and distribution of users' secret keys. Furthermore, AA is responsible for attribute revocation when users lost the authority to access the cloud data. CSS is responsible for storing the cloud data which is provided by DO and providing service for cloud access user.

When a new user joins the cloud storage system, AA generates the secret keys based on attribute sets of users. When users access their interested data, they update secret keys by using the master keys generated by DO. User can correctly decrypt the ciphertext only when attribute set related secret key satisfy the attribute policy. In our scheme, DO CSS and AA should stay online all the time, but the user doesn't have to stay online.

B. Security Model

The security of CP-ABE algorithm used in this paper is proved in the selective security model. In this paper, AA is fully trusted. We consider the honest but curious cloud service providers may try to obtain some valuable data files or the unauthorized users to steal the beneficial data through the data that is already in hand. Maybe there are collusion attacks of the two. Here we suppose the communication channels between data owners or users or cloud storage service providers and attributes authority presume to be security, and the DO, AA and CSS will not reveal the user's private keys nor utilize its privilege to decrypt the ciphertext.

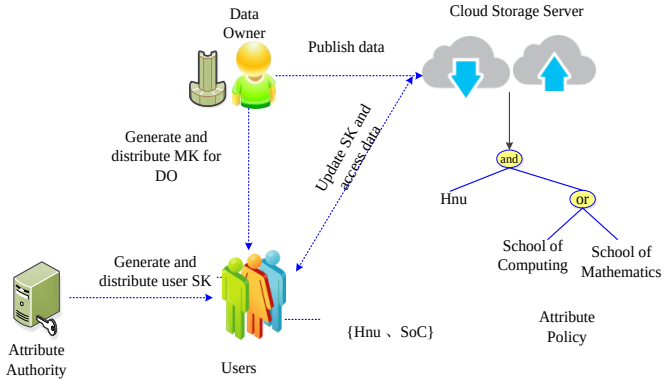

Fig.1 System Model

IV. ABE-WE FRAMEWORK

A. Design binary tree structure mechanism

Definition 4: complete binary tree is a structure tree which is composed of the token and random key. Each leaf node of binary tree contains a random key and each edge

connect node corresponds to a token. Furthermore, each leaf node and the corresponding user is one-to -one mapping.

Let $\oplus$ denotes the XOR gate and $H(\bullet)$ represents a public one-way hash function. The length of random key is decided by $H(\bullet)$. The binary tree is built with the following steps:

- The system generates a binary tree based on the total users;
- Randomly distribute a key (RK_i) to each node (n_i).
- As for the setting of the internal nodes, we adopt the bottom to top method. For the internal node we search the RK_i which belongs to the left child node, then the random key for the internal node n_j is generated by the hash function $RK_j=H(RK_i)$;
- As for the setting of token, let Flag=1 denote the hash function. If the edge is the connection between the parent node and the left child node, we set the token $TK_{i-j} = Flag$ .While for the right child node, let RK_i denote its random key and RK_j denotes its parent key. The token whose corrsponding edge is the connection between the parent node and the right child node is set to be $TK_{i-j}=H(RK_i) \oplus RK_j$.

Definition 5: Let MCKS denote Minimum Cover Key Set, and $MCKS_x \in MCKS(1 \le x \le k)$, let Φ_x be the user's set where all the users contain the attribute x, then Ψ_x represents the internal nodes set which covers the leaf nodes set Φ_x, $MCKS_x$ denotes the random keys set composed by all the random keys in Ψ_x. As depicted from fig.2, Φ_x denotes the leaf nodes set {n_1, n_2, n_5, n_6, n_7, n_8}, Ψ_x denotes the internal nodes set {n_9, n_{14}}, and $MCKS_x$ denotes the random keys of the internal nodes set { RK_9, RK_{14}}.

Definition 6: Let KCS denote Key Chain Set, n_i be a random node from binary tree, then KCS_i ($1 \le i \le m$) represents the set that contains all the random keys from the leaf node n_i to the root node. As depicted from fig.2, if the node is n_1, then KCS_1= { RK_1 ,RK_9,RK_{11},RK_{15} }.

Definition 7: Let TCS denote Token Chain Set, let n_i be a random leaf node, then TCS_i represents the set that contains all the tokens from the leaf node n_i to the root node. As depicted from fig.2, if the leaf node is n_2 ,then TCS_2={TK_{2-11}、Flag、Flag}.

Lemma 1: For a random user $u_i (i \in [1,m])$, let n_i be the mapping leaf node, if $u_i \in G(x)(x \in [1,k])$, then there is only one intersection between KCS_i corresponding to n_i and $MCKX_x$ corresponding to $G(x)$.

Proof: 1) Existence theorem: from the definition of KCS_i and $MCKX_x$, there is at least one intersection between KCS_i and $MCKX_x$ $(u_i \in G(x)(x \in [1,k]))$. 2) Uniqueness Property: if we assume that there are at least two intersections between KCS_i and $MCKX_x$, there are at least two nodes can cover n_i at the same time. It is inconsistent with the definition of $MCKX_x$, thus uniqueness is proved.

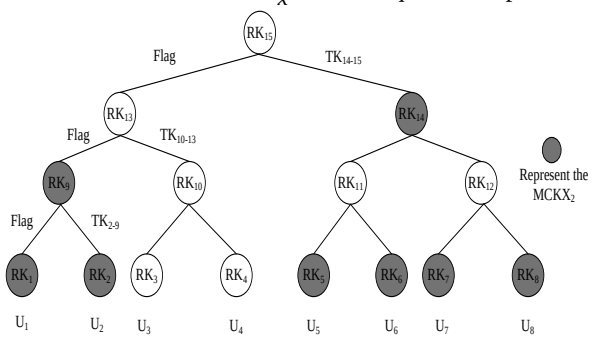

Fig.2 the binary tree structure mechanism

B. Implement of The Algorithm

Definition 8: attribute embeddoor, for a random attribute $x \in A$ we assign a unique embeddoor ED_x. ED_x can be obtained only when u_t satisfy $u_t \in G(x)$.

- Setup（m, k）: We fistly select the prime order p and generator *g* for multiplicative group G_0. Let *e* be the bilinear map $G_0 \times G_0 \to G_1$. Based on the attribute set $A = \{1,2,3,\ldots\ldots k\}$ and user set $U = \{u_1, u_2, u_3, \ldots\ldots, u_m\}$, we randomly choose $\eta_x, ED_x \in Z_p$, thus we obtain $E_x = g^{\eta_x ED_x}$, chosing $\beta \in Z_p$, generating $AMK =< \beta, \{ED_x\} >$, and the public key is published as APK= $< G_0, g, g^{\beta}, \{E_x\} >$, where $x \in A(1 \le x \le k)$. ED_x is used for computation of attribute vacation. Furthermore, randomly choose $\alpha \in Z_p$ and compute $OMK =< g^{\alpha} >$ for DO, and $OPK =< e(g,g)^{\alpha} >$ is published at the same time.
- KeyGen (AMK,S): Use the AMK to generate the private key corresponding to the attribute set S. Randomly choose $t \in Z_p$, compute $D = g^{\beta t}, L = g^t$. For a random attribute $x \in S$,compute
$$D_x = E_x^{t/ED_x} = g^{\eta_x t ED_x / ED_x} = g^{\eta_x t}$$
Randomly choose embeddoor EDKey, then the secret key is computed using the following equations:
$$SK =< D, L, \{D_x\}_{x \in S}, EDKey >$$
The variable t is used for the randomization of private key and resists the collusion attack; EDkey is used for recovering the attribute embeddoor.
- Encrypt（APK，OPK，P，m）: The encryption algorithm encrypts a message m by using APK and OPK, and attribute policy *P*. Where *P* represents the attribute policy (M, ρ). M is a matrix with the size defined as $\ell \times k$. ρ is an mapping function. Let $\vec{v} = (s, v_2, v_3 \cdots v_k)$ be a k dimensional vector

and $(s, v_2, v_3 \cdots v_k) \in Z_p$, compute $\tilde{C} = m \cdot e(g,g)^{\alpha s}$, $C = g^s$. Let $\vec{M}$ be a vector which is composed by ith row of matrix M and $i \in [1, \ell]$, then compute $\lambda_i = \vec{M}_i \cdot \vec{v}$, randomly choose $(r_1, r_2 \cdots r_\ell) \in Z_p$, compute $C_i = g^{\beta\lambda_i} E_{\rho(i)}^{-r_i}$, $C_i^{'} = g^{r_i}$ finally the cipertext is computed using the following equations:

$$CT = < (\mathrm{M}, \rho), \tilde{C}, C, C_i, C_{i(i \in [1,\ell])}^{'} >$$

- KeyUpdate (OMK, SK): Update the secret key to obtain the new SK.

$$SK = < D = g^{\alpha} g^{\beta t}, L, \{D_x\}_{x \in s}, EDKey >$$

- Decrypt（SK, CT）: We use the secret key SK to decrypt the ciphertext. Only when the related attribute of SK satisfies the attribute policy *P*, the ciphertext can be decrypted. According to the method proposed by Beimel A, we can compute a set of constant coefficients $\{\theta_i\}_{i \in I}$, $\sum_{i \in I} \theta_i \lambda_i = s$. thus we obtain A using the following equation:

$$\begin{aligned} A &= \prod_{i \in I} (e(C_i, L) e(C_i, D_{\rho(i)}^{ED_{\rho(i)}}))^{\theta_i} \\ &= \prod_{i \in I} (e(g^{\beta\lambda_i} g^{-r_i \eta_{\rho(i)} ED_{\rho(i)}}, g^t) e(g^{r_i}, g^{r_i t ED_{\rho(i)}}))^{\theta_i} \\ &= e(g,g)^{t\beta \sum_{i \in I} \theta_i \lambda_i} \\ &= e(g,g)^{t\beta s} \end{aligned}$$

To recover the plaintext:

$$\begin{aligned} &\tilde{C} / (e(C, D) / A) \\ &= m \cdot e(g,g)^{\alpha s} / (e(g^s, g^{\alpha} g^{\beta t}) / e(g,g)^{t\beta s}) \\ &= m \end{aligned}$$

C. Attribute Revocation

- The publication of the EDkey: As mentioned the relationship between leaf node and user is one-to-one mapping based on the mechanism of binary tree, and the corresponding random secret key is EDkey.
- For a random attribute $x \in A$ ($x \in [1,k]$), let $G(x)$ be the attribute group, we compute the minimum secret cover set $MCKS_x$, and also generate the embeddoor information EDM_x= $\{E_{RKj}(ED_x)\}$ where $RK_j \in MCKS_x$, E represents the fast encryption algorithm. ED_x represents the user's embeddoor. As depicted from fig. 2, we assume that $G(2) = \{u_3, u_4, u_5, u_6, u_7, u_8\}$ be attribute 2th attribute group, then we can obtain the $MCKS_2$=$\{RK_{10}$, $RK_{14}\}$ and the embeddoor information $EDM_2 = \{E(ED_{10}), E(ED_{14})\}$.
- When the users lose the access authority for the cloud data, AA is in charge of the user's attribute revocation. Let R be the attribute revocation set, u_t be the revocatory user. First, for a random attribute $x \in R$, we can update the attribute embeddoor ED_x to $ED_x^{'}$. Now $u_t \notin G(x)$, and we can obtain new $MCKS_x$. According to new G(x), we can update the EDM_x to new $EDM_x^{'}$= $\{E_{RKj}(ED_x^{'})\}$. Then, AA is used to update the AMK and ASK. For a random attribute $x \in R$, we can update the Ex which related to AMK to $E_x^{'} = g^{\eta_x ED_x^{'}}$ and replace the ED_x which related to ASK.
- DO executes the Lazy ciphertext re-encryption when the cloud data is modified. The new ciphertext CT' is obtained by Encrypt（AMK, OPK, *P*, m）.In this paper, we can make several times of attribute revocation to be one time re-encryption when the update of cloud data is not frequent. We can use Lazy ciphertext re-encryption on this condition. Thus, this scheme can effectively reduce the cost of re-encryption

V. SECURITY PROOF

Hypothesis 1: If we know the secret key for a random leaf node and the Token Chain Set (TCS), we can recover the Key Chain Set (KCS).

Proof: Randomly select a leaf node n_i, if n_i is a left child node, then the random secret key for the parent node n_j can be obtained by RK_j=H (RK_i), and we can obtain the KCS; if n_i is a right child node, then the random secret key for the parent node n_j can be obtained by RK_j=H (RK_i) $\oplus$ $TK_{i\text{-}j}$= H (RK_i) $\oplus$ H (RK_i) $\oplus$ RK_i.

Hypothesis 2: If the random secret key for a random leaf node is provided, even someone obtain all the tokens for a binary tree, he cannot recover the random secret key for all nodes other than the random secret key from the leaf node to the root node.

Proof: Let n_i be a random leaf node, we can obtain the secret key from n_i to root node based on the Hypothesis 1. Then, let n_j be a parent node and n_i to be a left child node, the right child node is n_j, we compute

$$\begin{aligned} &RK_k = RK_j \oplus TK_{k \to j} = RK_j \oplus H(RK_k) \oplus RK_j \\ &= H(RK_k) \end{aligned}$$

Because of the unidirectional property of hashing function, RK_k cannot be recovered. In a similar way, if n_i is be the right child node, we cannot recover the secret key neither.

The security of this scheme is depended on both the security of the cpabe-we encryption algorithm and the security of the attribute avocation scheme. The cpabe algorithm is proposed based on the PBDHE math problem, which is proved to be security under the standard model. cpabe-we algorithm is the inheritance and extension of cpabe algorithm from the following two aspects:

- Changing the generation of secret key, the ciphertext can be decrypted when user obtained the OSK which distributed by DO;
- The attribute embeddoor EDx is employed for each attribute x. Through the control of EDx, we can achieve fine-grained attribute revocation. EDx is encrypted by the random secret key and the binary tree mechanism is proved to be security from the Hypothesis 1 and Hypothesis 2. Thus the cpabe-we is also security under the standard model.

AA will randomly select the embeddoor when the attribute is revoked. Meanwhile, the revoked user neither can obtain the random key nor access the cloud data because of the new embeddoor. We can make a conclusion that capbe-we is security.

VI. CONCLUSION

How to guarantee the security of data under the cloud environment becomes a hot topic for recent years. This paper designs the cpabe with embeddoor encryption algorithm and the binary tree mechanism based on the existed cpabe. Take advantage of the design for embeddoor, it is unnecessary to update the non-revocatory user's secret key when dealing with the attribute revocation. Thus this scheme can effectively reduce the cost of re-encryption. Besides, the mechanism of binary tree can recover the random key from the leaf node to the root node through the tokens when user obtain the random key of leaf node, and it can also reduce the storage cost of random key. Our approach not only supports the data user from DO access the cloud data but also reduce the cost of computation and storage.

REFERENCES

[1] Larry D.Cloud computing hasn't gone fortune 500 yet,but it' scoming[EB/OL].(2007-12-18. http//blogs .zdnet.corn/BTL/? p=8199

[2] Cheristian C, Idik K,Shraer A.Trusting the cloud［J］.ACM SIGACT News,2009,40(2) :81-86.

[3] Shamir A.Identity-based cryptosystems and signature schemes[C] Advances in Cryptology CRYPTO'84.Berlin: Springer-Verlag, 1984: 47-53

[4] Boneh D, Franklin M.Indentify-based encryption from the weil pairing[C]// Advances in Cryptology-CRYPTO'01 .Berlin: Springer-Verlag, 2001:213-229

[5] Bethencourt J, Sahai A, Water B,Ciphertext-policy attribute based encryption[C]// Proc of IEEE Sympoisum on Security and Privacy.Washington DC:IEEE Computer Society,2007:321-334.

[6] Goyal V, Jain A, Pandey O, et al. Bounded ciphertext policy attribute based encryption[C]//Proc of ICALP. Berlin: Springer-Verlag, 2008:579-591

[7] Liang X H, Cao Z F, Lin H, et al.Provably secure and efficient bounded ciphertext policy attribute based encryption[C]//Proc of ASIAN ACM Sympoisum on Information, Computer and Communications Security. New York: ACM Press, 2009:343-352.

[8] Pirretti M, Traynor P, Mcdaniel P, Security attibute based systems[C]//Proc of ACM Conference on Computer and Communication Security. New York: ACM Press, 2006: 99-112

[9] Yu S C, Wang C, Ren K,et al. Attribute based data sharing with attribute revoation[C]// /Proc of ACM Conference on Computer and Communication Security. New York: ACM Press, 2010:261-270

[10] Hur J J，Noh D K. Attribute-based Access Control with Efficient Revocation in Data Outsourcing Systems[J].IEEE Trans on Parallel and Distributed Systems, 2011,22(7)：1214—1221.

[11] Bethencourt J, Sahai A, Waters B, Ciphertext-Policy attribute-based encryption. IN: Proc. of the 2007 IEEE Symp. on Security and rivacy. Washington: IEEE Computer Society 2007.321-334.[doi: 10.1109/SP.2007.1l]

[12] Hur J, Noh D K. Attribute-based access control with efficient revocation in data outsourcing systems [J]. Parallel and Distributed Systems, IEEE Transactions on, 2011, 22(7): 1214-1221.