A scheme of hidden-structure attribute-based encryption with multiple authorities

J Ling\(^1\) and A X Weng\(^{1*}\)

\(^1\)Faculty of Computer, Guangdong University of Technology, Guangzhou 510006, PR. China

E-mail:wenganxiang@163.com

Abstract. In the most of the CP-ABE schemes with hidden access structure, both all the user attributes and the key generation are managed by only one authority. The key generation efficiency will decrease as the number of user increases, and the data will encounter security issues as the only authority is attacked. We proposed a scheme of hidden-structure attribute-based encryption with multiple authorities, which introduces multiple semi-trusted attribute authorities, avoiding the threat even though one or more authorities are attacked. We also realized user revocation by managing a revocation list. Based on DBDH assumption, we proved that our scheme is of IND-CMA security. The analysis shows that our scheme improves the key generation efficiency.

1. Introduction

The cipher policy attribute-based encryption (CP-ABE)[1] has been used more and more widely since it was first proposed because of its flexible access control policy. Under CP-ABE, the encryptor uses public key to encrypt the message, and the decryptor gets decryption key from a trustable authority according to his attribute set. If the attribute set of decryptor is dissatisfied to the access structure embedded into the ciphertext, the decryptor cannot decrypt the ciphertext.

However, there are two main problems existing in the traditional CP-ABE schemes: (1) There is only one authority in most of CP-ABE schemes. The key generation efficiency will reduce as the number of attribute grows, and this will reduce the efficiency of the whole system. Besides, once the authority is attacked, the user’s key will be compromised, leading to the threat of user’s privacy and data security. (2) Both access structure and ciphertext are sent to the user in most of CP-ABE schemes, and the access structure probably includes information about the message. If the ciphertext is intercepted, the privacy of encryptor will be under the risk of being compromised.

To solved the first defect described above, Chase[2] first proposed multi-authority attribute-based encryption(MA-ABE). Gao et al.[3] introduced privacy homomorphism and proposed a scheme of MA-ABE for protecting user privacy. Li et al.[4, 5] proposed an adaptive secure ABE scheme with multiple authorities, and the attributes could be repeated. An ABE scheme was proposed by Wu[6], and it introduced proxy re-encryption to increase encryption efficiency and realized policy update. Neethu et al.[7] proposed a scheme, which deletes the central authority, and all the authorities generate secret key independently with each other. Besides, it enables users to decrypt more than one file by key aggregation concept.

To guarantee the confidentiality of access policy, Song et al.[8] used composite order bilinear groups to realize the policy-hidden scheme based on access tree. Based on policy-hidden ABE, Du et al.[9] and Lei et al.[10] proposed an access control scheme which is applicable in the cloud. Wang et al.[11] and Weng et al.[12] proposed the ABE schemes with hidden policy, which uses access tree to express access structure and realized anonymous access structure. Xie et al.[13] used asymmetric pairings to realize hidden access policy, but this scheme increases the computational overhead of bilinear pairings. Ying et al.[14] propose an ABE scheme, which supports dynamic policy update, but it the policy is only partially hidden.
The schemes mentioned above either adopted multiple authorities or realized hidden policy, but there exists rare scheme which combines these two features. Zhong et al. [15] realized hidden policy with multiple authorities as well as user revocation by the linear secret sharing scheme (LSSS), but neither encryption nor decryption are not efficient. Fan et al. [16] adopted key fragmenting to realize multi-authority attribute-based encryption with policy hidden, but the data owner needs to keep online. Aiming at the problems mentioned above, based on the scheme in [5], we propose a scheme of hidden-structure attribute-based encryption with multiple authorities, where the access structure is transformed to access tree, the user private key is generated by multiple authorities, increasing the key generation efficiency, and avoiding the security issues brought by the attacked authorities. We realized the hidden access structure and the user privacy could not be leaked by the attacked authorities with the condition that the authorities are not completely trustworthy. Besides, we allocated a global unique identifier, using which to generate a user revocation list which is used to update private key for legal users.

2. Preliminaries

2.1 Bilinear map. Let \( G_1 \) and \( G_2 \) be two multiplicative cyclic groups with prime order \( p \), and \( g \) be a generator of \( G_1 \). A map \( e : G_1 \times G_1 \to G_2 \) is a bilinear map if the following properties can be satisfied:

1. Bilinearity: \( \forall a, b \in \mathbb{Z}_p, \forall f, h \in G_1, e(fa, hb) = e(f, h)^{ab} \).
2. Non-degeneracy: \( \exists f \in G_1, e(f, f) \neq 1 \).
3. Computability: For all \( f, h \in G_1 \), there exists an algorithm to compute \( e(f, h) \).

2.2 Decisional bilinear Diffie-Hellman assumption (DBDH assumption). Choose some random values \( a, b, c, z \in \mathbb{Z}_p \). The DBDH assumption says that, there is no probabilistic polynomial time algorithm that can differentiate the tuples \( [g^a, g^b, g^c, e(g, g)^{abc}] \) and \( [g^a, g^b, g^c, e(g, g)^z] \).

2.3 Access Structure. We assume that there are \( n \) kinds of attributes in the system. Let \( U = \{ U_1, U_2, \ldots, U_n \} \) be the set of system attributes and \( U_i = \{ u_{i,1}, u_{i,2}, \ldots, u_{i,m} \} (m \geq 1) \), \( S = \{ S_1, S_2, \ldots, S_n \} \) be the set of user attributes, where \( S_i = \{ \text{att}_{i,1}, \text{att}_{i,2}, \ldots, \text{att}_{i,m} \} \). \( A = \{ A_1, A_2, \ldots, A_n \} \) indicates the access structure and \( A_i \subseteq U_i \). For \( \forall i \in [1, n] \), \( S_i \subseteq A_i \), we say that the user attributes satisfies the access structure.

In this paper, the access structure adopts the ‘AND-OR’ gates with multi-valued attributes. The access structure is transformed into an access tree before encrypting. The leaf node in the tree represents the attribute value, and the non-leaf node represents ‘AND’ gate or ‘OR’ gate. Each attribute name of leaf node is replaced by a value chosen by data owner when encrypting. A user will not obtain the information about the attributes and data owner. As is shown in the fig 1, ‘\( \wedge \)’ indicates the ‘AND’ gate, and ‘\( \lor \)’ indicates the ‘OR’ gate. Assuming that the access structure \( A = \{ \{ v_{1,2} \}, \{ v_{2,1}, v_{2,4} \}, \{ v_{3,1} \}, \{ v_{4,2}, v_{4,3}, v_{4,4} \} \} \) indicates the access structure, we say that the attribute set \( S = \{ v_{1,2}, v_{2,1}, v_{3,1}, v_{4,2} \} \) or \( S = \{ v_{1,2}, v_{2,4}, v_{3,1}, v_{4,3} \} \) satisfies the access structure.

Figure 1. Access structure
2.4 System composition. In this paper, there are five participants: cloud service provider (CSP), central authority (CA), attribute authority (AA), data owner (DO) and user (UID).

- **CSP**: Store encrypted data for DO, and it is not trusted.
- **CA**: Generate public parameters and confirm the legality of UID.
- **AA**: Each AA generates a part of private key for UID. In this paper, an AA can manage multiple attributes and each attribute is managed by only one AA.
- **DO**: Decide access structure and encrypt data, and upload the encrypted data to the cloud.
- **UID**: Download data from encrypted data. UID can decrypt the ciphertext if and only if its attribute set satisfies the access structure.

2.5 Security model. The scheme proposed in this paper satisfies indistinguishability of ciphertext under chosen-message (IND-CMA), the security model based on which is described as an interactive game between adversary A and challenger B as follows:

- **Step 1**: The adversary A submits the access structure $W$ to challenge to the challenger B. After selecting a security parameter $\mathcal{K}$, B runs initialization algorithm and get the system public key $PK$, the system main key $MK$, attribute authority public key ($APK$) and private key of attribute authority ($ASK$).
- **Step 2**: A queries B about attribute list $S$, but B cannot run the generation algorithm of private key and cannot send the private key $SK$ to A, unless $S$ does not satisfy the access structure $W$. A can query for polynomial times.
- **Step 3**: A submits two groups of plaintext $M_0$ and $M_1$, which are different but have the same length. B chooses $\varphi (\varphi \in \{0,1\})$ randomly by tossing a coin, then runs the encryption algorithm to encrypt $M_\varphi$, and returns $CT$ (the corresponding ciphertext of $M_\varphi$) to A.
- **Step 4**: Repeat step 2 and step 3, but the query from A must be limited, that is, $S$ does not satisfy $W$.
- **Step 5**: A outputs a guess $\varphi'$ of $\varphi$ ( $\varphi' \in \{0,1\}$). If $\varphi'=\varphi$, A wins the game, otherwise loses the game.

Suppose the advantage that A wins the game is $Ad_A= Pr[\varphi=\varphi']-\frac{1}{2}$.

Within polynomial time, if there is not a polynomial-time adversary who can win the game with an unnegligible advantage, we can say that the scheme of multiple-authority attribute-based encryption with hidden policy is of IND-CMA security.

3. Our scheme

(1) Initialization

**CA setup**: Input a security parameter $\mathcal{K}$, then output parameter as follows:

1) Generate a bilinear group $G$ and a bilinear map $G \times G \rightarrow G_T$, $G$ is a multiplicative cyclic group with prime order $N$.
2) CA selects $a \in Z_N^*$ randomly, and outputs the public key of system $PK = \{e, g, e(g, g)^a, G, G_T\}$ where $g$ is the generator of $G$.

The main key of system is $MK = \{g^a\}$

CA sends $MK$ to all the AAs through a secure channel.

**AA setup**: CA allocates a unique global identifier $aid$ for each legal AA. Let $U = \{U_1, U_2, \ldots, U_n\} = \{att_1, att_2, \ldots, att_n\}$ indicate a complete attribute set. Divide $U$ into $k (k \leq n)$ subsets and distribute these $k$ subsets to $k$ AAs to manage. AA randomly selects $a_{ij} \in Z_N^* (i \in [1,n], j \in [1,n_i])$, where $n_i$ represents the number of $att_i$. Each AA calculates its own public key $APK = \{g^{a_{ij}}\}$

And saves its own private key $ASK = \{g^a\}$.
ASK = \{a_{i,j}\}

**User registration:** CA allocates a unique global identifier \textit{uid} for each user, and uses \textit{uid} to generate identify code Certificate(\textit{uid}), which is stored in the database of CA. Certificate(\textit{uid}) is a hash algorithm which uses \textit{uid} to generate a hash code with fixed length.

(2) Key generation

After receiving decryption request from user, CA uses \textit{uid} to search Certificate(\textit{uid}) in database, to confirm the legality of this user. If this user is legal, CA will send a message to all the AAs that they can generate private key for this user, otherwise, CA will reject decryption request.

Let \(S_{\text{uid}}=\{\text{att}_{1,\text{uid}}, \text{att}_{2,\text{uid}}, \ldots, \text{att}_{m,\text{uid}}\}\) be the attribute set of UID, where \(S_{\text{uid}} \subseteq U, m \in [1, n]\). \(S_{\text{uid,aid}}=\{\text{att}_{1,\text{uid}}, \text{att}_{2,\text{uid}}, \ldots, \text{att}_{q,\text{aid}}\}\) represents the attributes of UID which AA\(_{\text{aid}}\) manages, where \(S_{\text{uid,aid}} \subseteq S_{\text{uid}}, q \in [1, m]\). AA\(_{\text{aid}}\) selects a value \(\lambda_{i} \in Z_{N}^{*}\) for each \(\text{att}_{i,\text{aid}}\), and uses MK sent by CA to calculate.

\[
D_{1} = g^{a_{i}} \cdot g^{\lambda_{i}a_{i,j}a_{j}} = g^{\lambda_{i}a_{i,j}a_{j}+a_{i}} \\
D_{2} = g^{\lambda_{i}}
\]

The output of the private key is

\[
SK = \{D_{1}, D_{2}\}
\]

(3) Encryption

1) Input the access structure \(A\), public key of the system(\(PK\)) and the plaintext(\(M\)).

2) The encryptor transforms \(A\) to the corresponding access tree \(\tau\) with the rule described in section 2.3. Let \(s\) be the root of \(\tau\), which is set to be the root node of the access tree and marked as read status. All the child nodes are marked as unread status. The encryptor conducts recursive computation for each unread non-leaf node.

Suppose \(t\) is the number of the child node of the current node. If current node is ‘\&’ and all of its child nodes are unread, the encryptor randomly chooses \(s_{i} \in Z_{N}^{*}\) for each child node, and calculates the value of the last leaf node as \(s_{j} = s - \sum_{i=1}^{t-1} s_{j} \mod N\), then sets these nodes as read status.

If current node is ‘\|’ and its all child nodes are under unread status, the encryptor randomly chooses \(s' \in Z_{N}^{*}\), assigns all the child nodes as \(s'\), then all the chilled nodes as read status. The ciphertext is calculated as follows.

\[
C_{0} = M \cdot e(g, g)^{a_{x}} \\
C_{1} = g^{s_{i}} \\
C_{2} = g^{s'_{i}}
\]

Finally, the output of the ciphertext is

\[
CT = \{C_{0}, C_{1}, C_{2}\}
\]

(4) Decryption

After the UID downloads ciphertext from CSP, UID can calculate as follows to decrypt if its attribute set satisfies the access structure.

\[
D' = \frac{\prod e(D_{1}, C_{1})}{\prod e(D_{2}, C_{2})} = \frac{\prod e(g^{a_{i}}, g^{\lambda_{i}a_{i,j}a_{j}})}{\prod e(g^{a_{i}}, g^{s_{i}})} = \frac{\prod e(g, g)^{\lambda_{i}a_{i,j}a_{j}+a_{i}}}{\prod e(g, g)^{s_{i}}} \\
= \frac{e(g, g)^{a_{x}} \cdot \prod e(g, g)^{\lambda_{i}a_{i,j}}}{\prod e(g, g)^{\lambda_{i}a_{i,j}}} = e(g, g)^{a_{x}}
\]
\[ M = \frac{C_0}{D^1} = \frac{M g(g,g)^{\lambda_2}}{e(g,g)^{\lambda_2}} \]

(5) User revocation
In this paper, the first step is that UID submits decryption request. If UID is legal, AA will allocate private key to UID. When CA finds malicious user, it will delete the corresponding Certificate\((uid)\) from its database, then generate a user revocation list, update \(MK\) and send them to all the AAs with secure channel. After receiving new \(MK\) and revocation list, AA will regenerate \(SK\) for the users who are not in the revocation list. The legality of malicious user cannot be confirmed when applying decrypt new file. Even though this illegal user jumps over the CA, it cannot get the new \(SK\) from AA.

4. Security proof

Based on DBDH assumption, our scheme is of IND-CMA security under the security model described in section 2.5. The proof is as follows.

Suppose \(G_1\) and \(G_2\) are two multiplicative cyclic groups with prime order \(p\), and \(g\) be a generator of \(G_1\). Map \( e: G_1 \times G_1 \rightarrow G_2 \) is a bilinear map. The challenger B tosses a coin to choose a number \( \xi \in \{0,1\} \).

If \( \xi = 0 \), B chooses \( (g, g^a, g^b, X) = (g, g^a, g^b, e(g,g)^\alpha) \), otherwise, B chooses \( (g, g^a, g^b, X) = (g, g^a, g^b, e(g,g)^\lambda) \), where \( a, b \in \mathbb{Z}_N \).

(1) The adversary A sends an access structure he wants to challenge to B. B calculates \( Y = e(g^a, g^b)^{x_1} \), then randomly chooses \( a, b \in \mathbb{Z}_N \) and executes the initialization algorithm to generate \( PK, MK \) and \( APK \), where \( PK = \{ e, g, g^a, g^b, G_{\Gamma} \} \), \( MK = \{ g^a \} \), \( APK = \{ g^{\alpha/\xi} \} \), \( ASK = \{ a, b \} \). B sends \( PK \) and \( APK \) to A and preserves \( MK \) and \( ASK \).

(2) A submits his attribute set \( S \) to B for private key query, but \( S \) does not satisfy \( W \). After receiving private key query, B randomly chooses \( \lambda, \xi \in \mathbb{Z}_N^* \) for each attribute of A, and calculates \( SK \) as follows.

\[ D_1 = g^{\alpha} \cdot g^{\lambda_{a,i}} = g^{\lambda_{a,i} + \alpha} \\
D_2 = g^{\lambda_{i}} \]

(3) B tosses a coin and outputs a random value \( b \in \{0,1\} \). Let \( \tau \) be the root of \( \tau \), which is set to be the root node of the access tree and marked as read status. All the child nodes are marked as unread status. B conducts recursive computation for each unread non-leaf node. Suppose \( v \) is the number of the child node of the current node. If current node is ‘\&’ and all of its child nodes are unread, B randomly chooses \( r_i \in \mathbb{Z}_N^* \) for each child node, and calculates the value of the last leaf node as \( r_v = \tau - \sum_{i=1}^{v-1} r_i \mod N \), then sets these nodes as read status.

If current node is ‘\vee’ and its all child nodes are under unread status, B randomly chooses \( r' \in \mathbb{Z}_N^* \), assigns all the child nodes as \( r' \), then all the chilled nodes as read status. The ciphertext is calculated as follows.

\[ C_0 = M \cdot e(g, g)^{\lambda_2} \\
C_1 = g^{\xi} \]

\[ C_2 = g^{r_{a,i}} \]

Finally, the output of the ciphertext is

\[ CT = \{ C_0, C_1, C_2 \} \]

B sends \( CT \) to A.

(4) Execute step(2).

(5) A outputs a guess \( \phi' \) of \( \phi \). If \( \phi' = \phi \), then B outputs the guess result \( \xi' = 0 \), otherwise, the guess result is \( \xi' = 1 \). So there are two situations as follows:
1) When $\xi' = 0$, $X = e(g^a, g^b) = e(g, g)^{\xi'}$, and the ciphertext is valid. A can guess the correct $\xi'$ with an unnegligible advantage. Under this condition, we have $\Pr[\varphi' = \varphi | \xi' = 0] = \frac{1}{2} + \epsilon$.

2) When $\xi' = 1$, $X = e(g, g)^z$. Because $z$ is a random number from $Z_N^*$, the ciphertext cannot be identified. Under this condition, we have $\Pr[\varphi' \neq \varphi | \xi' = 1] = \frac{1}{2}$.

Therefore, the probability that B successfully breaks the DBDH assumption is:

$$Ad = \frac{1}{2} (\Pr[\varphi' = \varphi | \xi' = 0]) + \frac{1}{2} (\Pr[\varphi' \neq \varphi | \xi' = 1]) - \frac{1}{2} = \frac{\epsilon}{2}$$

As we can know according to the proof above, If the advantage of A is $\epsilon$ within polynomial time, the probability that B breaks the scheme proposed in this paper is $\frac{\epsilon}{2}$. Therefore, if $\epsilon$ is unnegligible, the advantage that B breaks our scheme is also unnegligible.

5. Efficiency analysis

Let $|G|, |G_T|, |Z_N^*|$ respectively represent the bit length of $G, G_T, Z_N^*$. Let $t_G$, $t_{GT}$ be the time for calculating on the groups $G$ and $G_T$, and $t_b$ be the time for calculating bilinear pairs.

Let $n$ be the total amount of system attribute. $L = \sum n_i$ represents the total amount of possible attribute, where $n_i$ is the amount of attribute $i$.

Table 1. Comparison of parameter length

| Scheme       | PK          | MK          | SK          | CT          |
|--------------|-------------|-------------|-------------|-------------|
| [11]         | $(L+1)|G|+|G_T|$ | $L|Z_N^*|+|G_T|$ | $(2n+1)|G|$ | $(2n+1)|G|+|G_T|$ |
| [16]         | $|G|+|G_T|$ | $|Z_N^*|+|G|$ | $(2n+1)|G|$ | $(2n+1)|G|+|G_T|$ |
| Our scheme   | $|G|+|G_T|$ | $|Z_N^*|+|G|$ | $2n|G|$   | $2n|G|+|G_T|$ |

As is shown in table 1, both PK and MK have the same length as the scheme in [16], but are shorter than the scheme in [11]. For the reason that the DO does not take part in private key generation, so the length of private key and ciphertext are shorter than the scheme in [16].

Table 2. Comparison of time overhead

| Scheme       | Encryption   | Decryption   |
|--------------|--------------|--------------|
| [11]         | $(2n+1)t_G+t_{GT}$ | $(2n+1)t_b+3t_{GT}$ |
| [16]         | $(2n+1)t_G+t_{GT}$ | $(2n+2)t_b+4t_{GT}$ |
| Our scheme   | $2n*t_G+t_{GT}$   | $2n*t_b+2t_{GT}$   |

As can be seen from table 2, our scheme spends less time on both encryption and decryption than scheme in [11] as well as scheme in [16]. Because in the scheme in [16], DO participates generation of private key, our scheme spends more time. But for the scheme in [16], the cost for increasing efficiency of private key generation is the DO needs to keep online, which greatly consumes resources. Based on the above analysis, our scheme is better than both the scheme in [11] and the scheme in [16].

6. Conclusion

This paper improves the scheme of attribute-based encryption with only one authority, proposing a scheme of hidden-structure attribute-based encryption with multiple authorities. In our scheme, the private key of user is generated by multiple authorities, improving encryption efficiency as well as
eliminating the threat brought by one or more attacked authorities. Besides, we realized the user revocation by updating private key for legal users. We proved that our scheme is of IND-CMA security under DBDH assumption. Lastly, comparing with other schemes, we analyzed the efficiency of ours. But the prerequisite for realizing our scheme is that the center authority is completely credible. Once successfully attacks the center authority, the attacker can register any amount of legal user and get private key from the attribute authorities, so that the data security will be threaten. Our next research is multi-authority attribute-based encryption without center authority.

7. Acknowledgement

This work is supported by the science and technology project of Guangdong Province (No.2015B010128014, 2016B090918039, 2016B090918058), and the project of Guangzhou Science and Technology (No.201508010026, 201604016067).

References
[1] Bethencourt J, Sahai A and Waters B 2007 Ciphertext-Policy attribute-based encryption. Proc. of the 2007 IEEE Symp. on Security and Privacy. Washington: IEEE Computer Society.321-334.
[2] Chase M. 2007 Multi-authority attribute based encryption Theory of cryptography. Springer Berlin Heidelberg. 515-534.
[3] Gao A and Li Z Z 2013 Improving user’s privacy for multi-authority ABE using privacy homomorphism J. IEICE Transactions on Fundamentals of Electronics Communications and Computer Sciences. 96(3):724-727.
[4] Li Z H, Yang M M and Chen X Y 2015 Adaptive Secure and Attribute Repeatable Multi-authority ABE scheme J. Computer Engineering. 41(12):21-25.
[5] Li Z H and Chen X Y 2016 Research and application of multi-authority attribute-based encryption scheme J. COMPUTER ENGINEERING AND DESIGN. 37(8):1981-1985.
[6] Wu G Q 2016 Multi-Authority CP-ABE with Policy Update in Cloud Storage J. Journal of Computer Research and Development. 53(10):2393-2399.
[7] Neethu A N and Prem K 2016 Efficient Security in Decentralized Cipher text-Policy Attribute-Based Encryption. JISART. 2(5):682-686.
[8] Song Y, Han Z, Liu F M and Liu L 2015 Attribute-based encryption with hidden policies in the access tree J. Journal on Communications. 36(9):119-126.
[9] Du R Y, Shen J, Chen J and Zhou S G 2016 An Access Control Scheme in cloud Storage with Policy Hiding ABE J. Wuhan Univ.(Nat. Sci.Ed.). 62(3):242-248.
[10] Lei L, Cai Q W, Jing J W, Lin J Q, Wang Z and Chen B 2016 Enforcing access controls on encrypted cloud storage with policy hiding J. Journal of Software. 27(6):1432-1590.
[11] Wang H P and Zhao J J 2016 Ciphertext-policy Attribute-based Encryption with Anonymous Access Structure J. Computer Science. 43(2):175-178,198.
[12] Weng A X and Ling J 2017 Improved scheme of CP-ABE with Hidden Access Structure J. Computer Science. 44(11A):377-380.
[13] Xie L and Ren Y L 2015 Efficient attribute-based encryption with hidden access structures J. JOURNAL OF XIDIAN UNIVERSITY. 42(3):97-102.
[14] Ying Z B, Ma J F and Cui J T 2015 Partially policy hidden CP-ABE supporting dynamic policy updating J. Journal on Communication. 36(12):178-189.
[15] Zhong H, Zhu W L, Xu Y and Cui J 2016 Multi-authority attribute-based encryption access control scheme with policy hidden for cloud storage J. Soft Computing.1-9.
[16] Fan Y D and Wu XP 2017 A Multi-authority Attribute-based Encryption Scheme with Hidden Policies J. Computer Engineering. http://kns.cnki.net/kcms/detail/31.1289.TP.20170421.1439.010.html