Traceable and Complete Fine-Grained Revocable Multi-authority Attribute-Based Encryption Scheme in Social Network

Yanmei Li, Fang Qi, and Zhe Tang(✉)

School of Information Science and Engineering, Central South University,
Changsha 410083, China
{liyanmei_,csqifang,tz}@csu.edu.cn

Abstract. Nowadays the data and user information involved in the social network with the nature of high complexity. More and more service providers and users will share the data in the cloud servers. To keep the shared data confidential against untrusted third-party service providers, settle single point failure and solve performance bottlenecks of authorized center as will as secret key abuse of malicious users who disclose their own private key, we propose a multi-authority attribute-based encryption scheme which supports traceability and fine-grained revocation mechanism in social network. The scheme is based on traditional attribute based encryption, which can realize distributed access control and support complete fine-grained revocation mechanism. The security is proved in the standard model, which effectively solves the above problems.

Keywords: Attribute-based encryption · Revocable · Traceable
Fine-grained · Social network

1 Introduction

In recent years, the attribute-based encryption-ABE [1] mechanism has become a hotspot of cryptography. This scheme associate the user's private key and ciphertext with different attribute sets, respectively. When and only when the user's private key can satisfy the access policy in the ciphertext, the ciphertext can be decrypted, which can flexibly display the access policy and reduce the network bandwidth and computation cost of network node. Therefore, fine-grained access control can be widely used. In ABE scheme, since the user's private key is generated from attribute authority, the private key may be leaked by attribute authority or users and it is hard to judge who should be responsible for the leakage. Thus the problem of private key abuse is prominent. Attribute revocation is a useful way to solve above problem and also is a matter of research. According to the way of attribute revocation, there are two types: direct revocation and indirect revocation. Peng and Zhang [6] pointed out that, in indirect revocation,

G. Wang et al. (Eds.): SpaCCS 2017, LNCS 10656, pp. 87–92, 2017.
https://doi.org/10.1007/978-3-319-72389-1_8

only the users who are not revoked can update the secret key. The first proposed ABE scheme only supports threshold access control policy. To express a more flexible access control policy, researchers proposed two kinds ABE schemes: one is key-policy attribute-based encryption (KP-ABE) [2,3], in which user's private key associates with the access policy; the other is ciphertext-policy attribute-based encryption (CP-ABE) [4,5], in which the access policy in embedded in the ciphertext. Considering that the CP-ABE can resist collusion attack, guarantee the confidentiality and realize flexible access control. In this paper, we propose a traceable and complete fine-grained revocable multi-authority attribute-based encryption scheme. The contributions are as follows:

- Proposed scheme can reduce the risk of single point failure and bottleneck.
- The ciphertext can contain multiple revocation lists and the complete fine-grained control can be realized.
- We proposed the traceable algorithm to locate the malicious user, which can guarantee the security.

2 Proposed Scheme

In this section, we will introduce the traceable and complete fine-grained revocable multi-authority attribute-based encryption scheme in detail. The scheme contains the following eight algorithms:

GlobalInit. Suppose G, G_T are cyclic group with order $N = P_1P_2P_3$, where P_1, P_2, P_3 are distinct big prime numbers, e is the bilinear mapping $e : G \times G = G_T$. Suppose G_{P_1} is the subgroup of G with order $P_1.g$ is the generator of $G_{P_1}.G_{P_3}$ is the subgroup of G with order $P_3.Y$ is the generator of G_{P_3}. We can randomley select a subgroup h is G_{P_1} and two parameters a, α from Z_N. Input 1^τ where τ is the security parameter, this algorithm outputs the system public key:

$$GPAR = (N, g, h, Y, e(g, g)^a, g^a) \tag{1}$$

CASetup. This algorithm runs in dth authority CA_d input the system public key $GPAR$ and the index d of corresponding authority CA_d. CA_d randomly chooses $a_d, \alpha_d \in Z_N$ and runs the algorithm. Finally, the algorithm returns $CA_d's$ system public key and system master key as follows:

$$CAPK_d = e(g, g)^{\alpha_d} \tag{2}$$

$$CAMSK_d = (a_d, \alpha_d) \tag{3}$$

CA_d publishes the public key and keeps the private key, the revoke list T_d is initialized as an empty list.

AASetup. On input the system public key $GPAR$, the attibute domain U_k managed by attribute authority AA_k and the index k, suppose $att \in U_k$, randomly select $S_{att} \in_R Z_N$ and suppose $H_{att} = g^{S_{att}}$. This algorithm outputs the public key:

$$AAPK_k = (H_{att}) \tag{4}$$

of AA_k private key:

$$AASMK_k = (S_{att}) \tag{5}$$

and publishes the public key and keeps the private key.

Encrypt. Input the plaintext M, access structure $T = (A, \rho)$, system public key $GPAR$, the public key $CAPK_d$ of CA_d, public key $AAPK_k$ of AA_k. Finally the algorithm outputs the ciphertext CT. The access structure T is presented by a Lsss matrix (A, ρ), where A is $l \times n$ matrix. ρ will map every line A_x of matrix A into an attibute $\rho(x)$, where $x \in (1, 2, ..., n)$. ρ is not allowed to map the different line into a same attribute. Then, to build up a n-dimention vector $\overrightarrow{v} = (s, v_1, v_2, ..., v_{n-1})$, randomly select $n-1$ numbers $v_1, v_2, ...v_n - 1$ from Z_N and a secret s. Suppose $\lambda_x = A_x \cdot \overrightarrow{v}$. $R(x)$ is the revoke list of users and $S(x) = U - R(x)$ (suppose $S(x) \neq \emptyset$). If $S(x) \neq U$ and $R(x) \neq \emptyset$, randomly select $\gamma_x \in Z_N$ and compute $C_{x,1}, C_{x,2}$, where $C_{x,2}$ is the information of attribute revokation. The ciphertext CT is:

$$CT = \left\{ \begin{array}{l} C_0 = M \cdot e(g,g)^{s \cdot \alpha_d}, C_1 = g^{a_d}, \\ C_2 = g^s, C_3 = g^{\lambda_x}, \\ C_{x,0} = H_{att}^{\lambda_x}, C_{x,1} = g^{\gamma_x}, \\ C_{x,2} = (g^{\gamma_x} \cdot \prod\limits_{j \in S(x)} g_{n+1-j} \cdot g^d)^{\lambda_x}, \\ C_{x,3} = (\prod\limits_{j \in R(x)} g_{n+1-j})^{\lambda_x} \cdot g^{\alpha_d} \end{array} \right\} x \in \{1, 2, ..., l\} \tag{6}$$

CAKeyGen. By this algorithm, the user registers his/her identity information gid with the authorization center CA_d to obtain the their own secret key. When CA_d received the users identity gid, it randomly selects a number $r_{gid,d} \in Z_N$ and $r_{gid} = \prod\limits_{d=1}^{D} r_{gid,d}$, inputs system public key $GPAR$ and the private key $CAMSK_d$ of CA_d. This algorithm returns the users public key:

$$aucpk_{gid,d} = \{g^{\alpha_d, g^{a \cdot r_{gid,d}}}\} \tag{7}$$

and users private key:

$$aucsk_{gid,d} = \{d, gid, L_{gid,d=g^{r_{gid,d}}}\} d \in \{1, 2, ..., D\} \tag{8}$$

AAKeyGen. This algorithm is used to generate the users attribute private key. When attribute authority receives the private key generation request from attribute $att(att \in U_k)$, input the system public key $\{aucpk_{gid,d}\}_{d \in D}$. Attribute authority AA_k computes CA_d's user's attribute private key:

$$auask_{att,gid,d} = (L_{gid,d})^{S_{att}} = g^{r_{gid,d} S_{att}} = (H_{att}^{gid,d}) \tag{9}$$

then combined with the user attribute private key of all authorization centers CA_d, we can get user's private key is:

$$auask_{att,gid} = \prod_{d=1}^{D} auask_{att,gid,d} = (L_{gid})^{S_{att}} \tag{10}$$

It is clear that the users identity is embedded in the private key.

Decrypt. If the visitor can satisfy the access structure $T = (A_{l \times n}, \rho)$, then there exists recovery parameter $\omega_x \in Z_N$ which can satisfy $\Sigma_{\rho(x) \in A} \omega_x \cdot A_x = (1, 0, ..., 0)$. If the visitor cannot satisfy the access structure $T = (A_{l \times n}, \rho)$, the algorithm return $\perp$, which means visitor cannot decrypt the ciphertext. D_x can be get from:

$$\begin{aligned} &\frac{e\left(g^{\alpha} g^{gid} uask_{gid,d}, C_3\right) e\left(C_3, C_{x,3}\right) e\left(g, \left(\prod_{j \in S(x)} g_{n+1-j}\right)^{\lambda_x}\right) \cdot e\left(C_3, C_{x,1}\right)}{e\left(L_{gid,d}, (H_{att})^{\lambda_s}\right) e\left(g, C_{x,2}\right) e\left(g, \left(\prod_{j \in R(x)} g_{n+1-j}\right)^{\lambda_x \gamma_x}\right)} \\ &= e(g,g)^{(gid+\alpha_c)\lambda_x} \end{aligned} \tag{11}$$

Suppose ω_x is the recovery parameter of the xth line of matrix A. According to LSSS recovery algorithm, the plaintext M can be recobered:

$$C_0 \cdot \left(\prod_{x \in l} D_x^{\omega_x}\right)^{-1} \cdot e(g^{gid}, g^s) = M \tag{12}$$

Trace. This algorithm is run by authority, because the attribute and identity is bond to the secret key. The algorithm will generate the pirate decoder, and then the identity of malicious user can be traced according to T.

3 Security Analysis

In this section, we will analysis the security performance. First, we will build semi-function ciphertext and semi-function secret key. Semi-function ciphertextrandomly select two numbers $c, t \in Z_N$, suppose R_x is the subgroup of G_{P_2}, g_2 is the generator of G_{P_2}. Randomly select a vector $\overrightarrow{u} = (c, u_2, u_3, ..., u_k)$ from Z_N^k. The semi-function ciphertext is CT. Semi-function secret key: there are two kinds semi-function secret keyrandomly select a number $n \in Z_N$. The first kind of semi-function secret key is:

$$\begin{aligned} aucsk'_{gid,d} &= \{d, gid, L_{gid,d} = g^{r_{gid,d}} g_2^{n + \overrightarrow{A}_x \cdot \overrightarrow{u}}\} \\ uask_{att,gid} &= \prod_{d=1}^{D} auask_{att,gid,d} = (L - gid^{S_{att}}) g_2^{n + \overrightarrow{A}_x \cdot \overrightarrow{u}} \end{aligned} \tag{13}$$

The second kind of semi-function secret key is:

$$\begin{aligned} aucsk'_{gid,d} &= \{d, gid, L_{gid,d} = g^{r_{gid,d}} g_2^{n}\} \\ uask_{att,gid} &= \prod_{d=1}^{D} auask_{att,gid,d} = (L - gid^{S_{att}}) g_2^{n + \overrightarrow{A}_x \cdot \overrightarrow{u}} \end{aligned} \tag{14}$$

For any $1 \leq k \leq q$ we define a serial security game, where q is the maximum number that adversary queries ciphertext:

- Game$_{real}$: The game is a real attack game defined earlier in this article.
- Game$_0$: This game is like $Game_{real}$, the difference is that challenger returns semi-function cipertext, so the game equals to $Game_{0,2}$.
- Game$_{final}$: In the challenge phase of the security game, the challenger returns a semi-function ciphertext with random message to the adversary.

Lemma 1. *If there is a polynomial time algorithm adversary Γ that $Game_{real}$ and $Game_0$ can be distinguished by non-ignorable advantage ε, then we can construct a polynomial time algorithm δ that can overcome the assumption 1 by advantage ε.*

Proof. The polynomial time algorithm receives a hypothetical 1 with condition (g, Y, T).

Init: The adversary outputs an access structure $T^* = (A^*_{l \times n}, \rho)$ that is challenged, for any $gid \in GID$, arbitrarily lists the specified user revocation list $R^*_{\rho(x)}$, and the algorithm δ runs Γ.

Setup: Suppose $\omega^* = \{\rho(x)\}$, generate the system public key:

$$GPAR = (N, g, h, Y, e(g, g)^{\alpha}, g^{a}) \tag{15}$$

System private key $CAMSK_d = (a_d, \alpha_d)$ and $AASMK_k = (S_{att})$.

Phase 1: When adversary asks about the private key rival, algorithm can use the system private key and secret key generation algorithm, then returns normal private key to adversary.

Challenge: Adversary submits two plain messages M_0, M_1 with the same length-randomly select a vector $\overrightarrow{v'} = (1, v'_1, v'_2, ..., v'_n)$, then choose a message M_θ from M_0, M_1, the ciphertext is:

$$CT' = \left\{ \begin{array}{l} {C_0}' = M_\theta \cdot e(g,g)^{s \cdot \alpha_d}, {C_1}' = g^{a_d} g_2^{c}, \\ {C_2}' = g^{s} g_2^{t}, {C_3}' = g^{A_x \cdot \overrightarrow{v'}} g_2^{ct}, \\ {C'}_{x,0} = {H_{att}}^{A_x \cdot \overrightarrow{v'}}, {C_{x,1}}' = g^{\gamma_x}, \\ {C_{x,2}}' = \left(g^{\gamma_x} \prod\limits_{j \in S(x)} g_{n+1-j} \cdot g^{d} \right)^{A_x \cdot \overrightarrow{v'}}, \\ {C_{x,3}}' = \left(\prod\limits_{j \in R(x)} g_{n+1-j} \right)^{A_x \cdot \overrightarrow{v'}} \cdot g^{\alpha_d} \end{array} \right\} x \in \{1, 2, ..., l\} \tag{16}$$

Phase 2: Repeat Phase1s operation. Guess: Adversary outputs the guess θ' of ciphertext M_θ.

- If $T = g^{s} \in G_{P_1}$ the query ciphertext is a normal ciphertext, the $Game_{real}$ game is running:

- If $T = g^s X^c \in G_{P_1 P_2}$,then $\overrightarrow{u} = c \cdot \overrightarrow{v'}$, the query ciphertext is a semi-function ciphertext, and the $Game_0$ game is running.

Therefore, the algorithm δ can overcome the hypothesis 1 with the advantage ε according to the adversary's output.

4 Conclusion

In this paper we proved that the scheme is secure in view of the problems of single authorization center and coarse-graind control in social network. This scheme not only solves the single node failure and performance bottlenecks of the authorization center, but also solves the problem of private key leakage by malicious users and achieves the completely fine-grained revocation. In fact, there are other aspects of revocation, such as the multi-authority CP-ABE [7], cyclical update of the user's private key [8] and other cancellation programs. Future direction of our research is how to generate a more secure and efficient solution based on existing works.

Acknowledgments. This work is supported by the National Natural Science Fundation of China under Grants No. 61632009 and the Science and Technology Project of Changsha under Grant No. kq1701089, and Fundamental Research Funds of Central South University under Grant No. 2017zzts713.

References

1. Bethencourt, J., Sahai, A., Waters, B.: Ciphertext-policy attribute-based encryptio. In: IEEE Symposium on Security and Privacy, pp. 321–334. IEEE Computer Society (2007)
2. Rahulamathavan, Y., Veluru, S., Han, J., et al.: User collusion avoidance scheme for privacy-preserving decentralized key-policy attribute-based encryption. IEEE Trans. Comput. **65**(9), 2939–2946 (2016)
3. Attrapadung, N., Libert, B., de Panafieu, E.: Expressive key-policy attribute-based encryption with constant-size ciphertexts. In: Catalano, D., Fazio, N., Gennaro, R., Nicolosi, A. (eds.) PKC 2011. LNCS, vol. 6571, pp. 90–108. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-19379-8_6
4. Wang, H.P., Zhao, J.J.: Ciphertext-policy attribute-based encryption with anonymous access structure. Comput. Sci. 2016
5. Ning, J., Dong, X., Cao, Z., et al.: White-box traceable ciphertext-policy attribute-based encryption supporting flexible attributes. IEEE Trans. Inf. Forensics Secur. **10**(6), 1274–1288 (2015)
6. Peng, K., Zhang, X.: Adaptively security CP-ABE scheme supporting attribute revocation. Comput. Eng. **41**(4), 151–155 (2015)
7. Li, Q., Xiong, J., Xiong, J., et al.: Provably secure unbounded multi-authority ciphertext-policy attribute-based encryption. Secur. Commun. Netw. **8**(18), 4098–4109 (2015)
8. Phan, D.H., Trinh, V.C.: Identity-based trace and revoke schemes. In: Boyen, X., Chen, X. (eds.) ProvSec 2011. LNCS, vol. 6980, pp. 204–221. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-24316-5_15