Location Privacy Preserving Scheme Based on Attribute Encryption

Xi Lin, Yiliang Han(✉), Yan Ke, and Xiaoyuan Yang

Department of Electronic Technology, Engineering University of Chinese Armed Police Force, Xi'an 710086, China
hanyil@163.com

Abstract. There are only two modes, "public" or "private" to manage the user's location information in the social network. However, in some cases, users need to inform some people of their exact location, and the other people are only access to the inaccurate, maybe fuzzy location information. Therefore, we design a location privacy preserving scheme based on attribute encryption, which provides "precise", "more accurate", "fuzzy" and "private" four modes to manage the location information. The scheme based on the algorithm of WT-CP-ABE [1]. The location information is divided into three parts according to different ranks of intimacy, then we encrypt the key information and position information with attribute-based encryption and symmetric encryption respectively, and then issue the ciphertext to the social network. We analyze the security of the scheme, which shows that the scheme has the advantages of user attribute information confidentiality, data confidentiality and it can resist the collusion attack.

Keywords: Social network · Location · Privacy protection · Ranks of intimacy · Attribute-based encryption

1 Introduction

Mobile social networking is an online platform built on some certain social relations. In mobile social network, users can share their interests, hobbies, status and daily activities with friends and families to strengthen the contact each other and maintain the deep affection. Users can also use positioning technology on mobile phone or other intelligent devices to share their location information and get some sorts of location-based services (LBS).

However, as people enjoy the convenience brought by the positioning technology, personal location privacy is suffering serious threats [2]. To the protection of personal location privacy, there are mainly two kinds of methods, spatial cloaking and space twist [3, 4]. Location k-anonymity model [5] is the most commonly used among the technology of spatial cloaking. When users need to provide personal location information, it will collect and send location information of k users in a large enough area (hereinafter referred to as fuzzy area) to the Service Provider (SP). As a result, the server can't distinguish the location of the users'. Space twist is going to generate some false positions and then both the real positions and false positions will be sent to the

G. Wang et al. (Eds.): SpaCCS 2016, LNCS 10066, pp. 164–177, 2016.
DOI: 10.1007/978-3-319-49148-6_15

server at the same time, therefore the user's location information will be hidden [3]. However, all these location privacy protection methods above assume the LBS provider as an attacker, and none of them think of the fact that the attacker may also be the user's "friends".

In fact, not all "friends" are credible and there may also be a potential attacker in the user's "friends list" [6]. Therefore, if the user publishes precise location indifferently to all friends in the network, it will inevitably cause the leakage of personal privacy and thereby users may suffer from security threats. In fact, when users post status, most of the social software, such as WeChat, Twitter provide the "visible" option, but they actually provide only "public" and "private" two options, therefore users are unable to show their precise location information to some close friends while showing fuzzy or not accurate location information to some good friends. Therefore, our paper contributes a location privacy preserving scheme based on attribute encryption. In our scheme, mobile social network users can choose "precise", "more accurate", "fuzzy" and "private" four modes for every friends according to the different ranks of intimacy, and friends in different ranks of intimacy can see different information of position. Besides, this scheme supports revocation of users' attributes, it can resist collusion attack and it is confidential in both users' attribute information and data.

2 Preliminary

2.1 Bilinear Maps

G1, G2 are two multiplicative cyclic groups of prime order p. If they satisfy the properties: (1) Bilinearity. For $\forall u, v \in G1$ and $\forall a, b \in Zp$, we have $e(u^a, v^b) = e(u, v)^{ab}$. (2) Non-degeneracy. $\exists u, v \in G1$, let $e(u, v) \neq 1$. (3) Calculability. $\forall u, v \in G1$, we can figure out $e(u, v)$ in a polynomial time. We call map e: $G1 \times G1 \rightarrow G2$ as the bilinear map [7].

2.2 Ciphertext-Policy Attribute-Based Encryption (CP-ABE)

In our paper, we use ciphertext-policy attribute-based (CP-ABE) to encrypt the information of key. We define A as the set of all the attributes {1, 2, …, k}, S as the non-empty subset of A. P is the attribute strategy contributed by "AND" and "OR". The commonly used policy of CP-ABE to control the access is based on the access to the tree structure [8] or linear secret sharing [9, 10]. In our paper, we use linear secret sharing to make it and the concrete process is described as follows: (1) We use (M, ρ) to express the property strategy of P. M is a matrix of $l \times h$ and ρ is a one-way function. When $i = 1, \cdots, l$, ρ(i) represents the ith line associated attributes of M. (2) When the set S satisfies the attribute strategy P, $I = \{i \mid P(i) \in S\}$, then we can thus calculate a constant coefficient group $\{\theta i \in Zp\}$ satisfying $\sum_{i \in I} \theta_i \vec{M}_i = \{1, 0, \ldots, 0\}$, and $\vec{M}_i$ is ith line vector of M. If S doesn't satisfy the attribute strategy P, there are no such group of constant coefficients. (3) Share the secret. We assume that $s \in Zp$ is a secret need to share, then we randomly select $h - 1$ value, $v2, v3, \cdots, vh \in Zp$,

contributing a vector of h dimensions $\vec{v} = (s, v_2, \cdots, v_h)$, and then we calculate $\lambda_i = \vec{M}_i \vec{v}$(i $= 1, 2, \ldots,$ l), in which λ_i is the value of sharing secret. Only when the attribute set S satisfies the attribute strategy P can we figure out the secret $s = \sum_{i \in I} \theta_i \lambda_i$.

2.3 The Mechanism of Token Tree

A tree of tokens. Our scheme constructs a complete binary tree as a token tree whose depth is D. The maximum of nodes on each $1 \sim$D-1 floor is 2^{D-1}, and nodes in D are concentrated in the far left. Each edge of tree corresponds to a token, each node has the corresponding random key, and each leaf node corresponds to a user in the system. We define Φ_x as a set of the leaf nodes in token tree corresponding to users in attribute group G(x) and define Ψ_x as the minimum set of nodes which could cover Φ_x, then we call the set of all the random key corresponding to all the nodes in set Ψ_x as the minimum covering key set (MCKS) of attribute x, written as $MCKS_x$. If n_i presents a leaf node in the token tree, our scheme defines the set of all the random key corresponding to the nodes from n_i to the root node, including the root node and leaf nodes as the key chain set (KCS) of node n_i, written as KCS_i and defines the set of all the tokens it gets through from n_i to the root node as token chain set (TCS), written as TCS_i.

The mechanism of the token tree. Our scheme makes each leaf nodes in the token tree corresponds to a user u_t in the system and takes the random key of the leaf nodes as TDKey in users' private key. The security of the token mechanism in our scheme depends on three theorems in the token trees: (1) If you know the random key corresponds to the leaf node n_i and all tokens of all the edges it gets through on its way to the root node, then we can figure out the keychain set KCS_i corresponds to n_i. (2) If we only know the random key corresponds to the leaf node n_i, we can`t get any random key corresponds to the nodes except for the nodes which get through from n_i to root node even if we get all the tokens in the token tree. (3) If the n_t represents the leaf node corresponds to user $u_t (1 \leq$ t$\leq$ m), then there is only one element makes the KCS_i of n_t intersect the $MCKS_x$ of G(x) when $u_t \in$ G(x) $(1 \leq$ x$\leq$ k).

3 Our Construction

3.1 System Model

Privacy protection social network system (PPSNS) is shown in Fig. 1. Just like literature [11, 12], we assume attribute authority AA is credible, and AA is responsible for initialization of the system, generation and distribution of user private key and management of users' attributes. Social network service provider SNSP is responsible for storing the information of the location released by data owner DO and providing users with social network service. Data owner DO is responsible for generating and distributing his own master private key and designing the strategy of encryption for data. When visitors want to get access to the DO`s data, they need DO`s own master private key to update their private keys. If and only if the visitor's attributes satisfy the attribute strategy of encryption can visitors decrypt the data correctly.

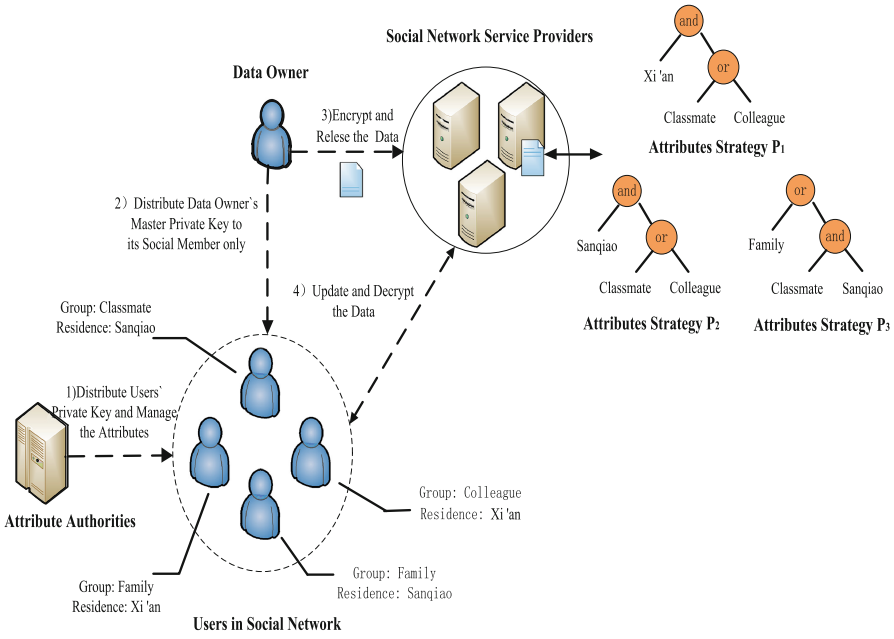

Fig. 1. Privacy protection social network system

3.2 Definitions

Definition 1. Attribute group. Users set U $= \{u_1, \cdots, u_m\}$, attribute set A = {1, 2, ⋯, k}, then we define the attribute group G(x) is a set of all the users who have the attribute of x.

Definition 2. Attribute trapdoor. If there is always an attribute trapdoor for each attribute x ∈ A, if and only if the user $u_t \in G(x)$ can the user obtain the attribute trapdoor for attribute x.

Definition 3. Location information. We divide the location information m of DO into m_1, m_2, m_3, for example, divide m = "Qitian hotel, town of Sanqiao, Xi'an, Shaanxi province" into m_1 = "Xi'an, Shaanxi province", m_2 = "town of Sanqiao", m_3 = " Qitian hotel".

Definition 4. Rank of intimacy. According to user's relationship with the DO (strangers, ordinary, friendly and intimate), we can divide DO's friends into four ranks of intimacy, written as 0,1,2,3. And we design three kinds of attribute strategy P_1, P_2, P_3. If the user is in the rank of the "ordinary", then set the user's rank = 1 and user's attributes set S satisfies strategy P_1, If "friendly", then set rank = 2 and S satisfies strategy P_2, and if "intimate", then set rank = 3 and S satisfies strategy P_3.

3.3 Our Algorithm

In this paper, we divide data needs to encrypt into two parts, location information and key information. Then we use traditional symmetric encryption to encrypt the location information and use WT-CP-ABE [1] encryption algorithm to encrypt key information. WT-CP-ABE algorithm is based on CP-ABE. WT-CP-ABE let AA and DO complete the generation of key together, and add attribute trapdoors and attribute revocation function to CP-ABE. Our algorithm includes Setup (), KeyGen (), EncryptM (), EncryptK (), KeyUpdate (), DecryptK () and DecryptM () seven sub function.

(1) Setup (1^{λ})
According to a given security parameter 1^{λ}, AA choose a multiplicative group whose order is p and generator is g, then there is a bilinear mapping $e : G_1 \times G_1 \rightarrow G_2$. We define attribute set in the system A = {1, 2, ⋯, k} and choose η_x and $TD_x \in Z_p$ randomly for attributes x ∈ A $(1 \leq x \leq k)$, then we calculate $T_x = g^{\eta_x TD_x}$. Randomly select $\beta \in Z_p$, generate system master private key ASK = <β, $\{TD_x\}$>, and issue public key APK = <G_1, g, g^{β}, $\{T_x\}_{x \in A}$>. Finally, DO select $\alpha \in Z_p$ randomly, calculate users private key OSK = <g^{α}> and issue the users' public key OPK = <$e(g,g)^{\alpha}$>.

(2) KeyGen (ASK, S)
Use master private key ASK to generate private key corresponding to the attribute set. Randomly choose $t \in Z_p$, calculate $D = g^{\beta t}$, $L = g^{t}$. Calculate D_x = = $g^{\eta_x t}$ for any attribute x ∈ S. Randomly select the trapdoor key TDKey, making up the user's private key SK = <D, L, $\{D_x\}_{x \in S}$, TDKey> . In order to avoid collusion attack, we add a random number t to D_x to randomize user's private key. Using the trapdoor key TDKey, we can restore the corresponding attributes trapdoor.

(3) EncryptM (m, k_1, k_2, k_3)
Use symmetric encryption to encrypt location information m. First of all, randomly choose three randomly generated symmetric key k_1, k_2, k_3, then divide location information m into m_1, m_2, m_3 according to the definitions in Sect. 3.2 and use k_1, k_2, k_3 to encrypt m_1, m_2, m_3. Then, we can get $E_{k_1}(m_1)$, $E_{k_2}(m_2)$ and $E_{k_3}(m_3)$. Output the cipher of location information CM :

$$CM = \langle E_{k_1}(m_1), E_{k_2}(m_2), E_{k_3}(m_3) \rangle$$

(4) EncryptK (APK, OPK, P_{rank}, K_{rank})
Use public key APK, OPK and attribute strategy P_{rank} to encrypt the information of symmetrical key K_{rank}. First of all, DO can design three attribute strategies P_1, P_2, P_3 according to the "ordinary", "friendly", "intimate" three ranks of intimacy. According to [9], we can get (M, ρ) representing the attribute strategy P_{rank}, rank ∈ {1, 2, 3}. M is a matrix of l × h, ρ is a one-way function. Then randomly select a vector of h dimensions $\vec{v} = (s, v_2, \cdots, v_h) \in Z_p$ and calculate $\tilde{C} = K_{rank} \cdot e(g,g)^{\alpha s}, C = g^{s}$. For any i ∈ {1, 2, ⋯, l}, let $\vec{M}_i$ be the ith row vector of M and calculate $\lambda_i = \vec{M}_i \vec{v}$. Then, choose random numbers $r_1, \cdots, r_l \in Z_p$ and calculate $C_i = g^{\beta \lambda_i} T_{\rho(i)}^{-r_i}$, $C_i' = g^{r_i}$, Output the ciphertext CK_{rank}:

$$CK_{rank} = \left\langle (M, \rho), \tilde{C}, C, \{C_i, C_i'\}_{i\{1,2,\cdots,l\}} \right\rangle$$

In the ciphertext, each C_i can be figured out through $T_{\rho(i)}$, but only get the attribute trapdoor $TD_{\rho(i)}$ can we figure out $T_{\rho(i)}$. Therefore, if and only if get the attributes trapdoors $TD_{\rho(i)}$ can we get C_i and decrypt the ciphertext.

When rank = 1, 2, 3, we use attribute strategies P_1, P_2, P_3 to complete encryption. K_1, K_2 *and* K_3 represent k_1, $k_1 \,||\, k_2$ and $k_1 \parallel k_2 \parallel k_3$ respectively. Then we can get CK_1, CK_2, CK_3 after the calculation, making up ciphertext of key:

$$\mathrm{CK} = \langle CK_1, CK_2, CK_3 \rangle$$

(5) KeyUpdate (OSK, SK)
Use OSK to update private key SK. Get new *SK*:

$$SK = \left\langle D = g^{\alpha} g^{\beta t}, L, \{D_x\}_{x \in S}, TDKey \right\rangle$$

(6) DecryptK (SK, CK)
Decrypt the ciphertext of key CK_{rank}. If and only if the attribute set S of private key SK satisfies the attribute strategy P_{rank} adopted in encryption of CK_{rank} can users decrypt the ciphertext and get the key CK_{rank}. Let $\mathrm{I} = \{\mathrm{i} \mid \rho(\mathrm{i}) \in \mathrm{S}\}, \mathrm{W} = \{\rho(\mathrm{i}) | \rho(\mathrm{i}) \in \mathrm{S}\}$ and assume that we has figure out the attributes trapdoor $TD_{\rho(i)}$ for each attribute $\rho(\mathrm{i}) \in \mathrm{W}$ by using the trapdoor key TDKey.

First, use the method referred in literature [9] to figure out a set of constant coefficients $\{\theta_i\}_{i \in I}$ and let the $\sum_{i \in I} \theta_i \lambda_i = s$. Then, calculate

$$A = \prod\nolimits_{i \in I} \left(e(C_i, L) e(C_i', D_{\rho(i)}^{TD_{\rho(i)}})\right)^{\theta_i}$$

$$= \prod\nolimits_{i \in I} \left(e(g^{\beta \lambda_i} g^{-r_i \eta_{\rho(i)} TD_{\rho(i)}}, g^t) e(g^{r_i}, g^{\eta_{\rho(i)} t TD_{\rho(i)}})\right)^{\theta_i}$$

$$= e(g, g)^{t\beta \sum_{i \in I} \lambda_i \theta_i}$$

$$= e(g, g)^{t\beta s}$$

Finally, we can get

$$K_{rank} = \tilde{C} / (e(C, D)/A)$$

$$= \tilde{C} / (e(g^s, g^{\alpha} g^{\beta t}) / e(g, g)^{t\beta s})$$

If the attribute set S of private key SK doesn't satisfy the strategies P_1, P_2, P_3 represented by (M, ρ) in CK_1, CK_2 or CK_3, we can't get K_{rank}; If the attribute set S of private key SK satisfies the strategy P_1 represented by (M, ρ) in CK_1, we can get K_1; If the attribute set S of private key SK satisfies the strategy P_2 represented

by (M, ρ) in CK_2, we can get K_2; If the attribute set S of private key SK satisfies the strategy P_3 represented by (M, ρ) in CK_3, we can get K_3.

(7) DecryptM (K_{rank}, CM)
Use the symmetric key k_1, k_2, k_3 we get from K_{rank} to decrypt cipher CM. When we get K_1, due to $K_1 = k_1$, we can use k_1 to decrypt $E_{k_1}(m_1)$ in CM. $D_{k_1}(E_{k_1}(m_1))$ and we can get fuzzy location information m_1; When we get K_2, due to $K_2 = k_1 || k_2$, we can use k_1, k_2 to decrypt $E_{k_1}(m_1)$ and $E_{k_2}(m_2)$ in CM. $D_{k_1}(E_{k_1}(m_1))$, $D_{k_2}(E_{k_2}(m_2))$ and we can get more accurate location information $m_1 \parallel m_2$; When we get K_3, due to $K_3 = k_1 \parallel k_2 \parallel k_3$, we can use k_1, k_2, k_3 to decrypt $E_{k_1}(m_1)$, $E_{k_2}(m_2)$ and $E_{k_3}(m_3)$ in CM. $D_{k_1}(E_{k_1}(m_1))$, $D_{k_2}(E_{k_2}(m_2))$, $D_{k_3}(E_{k_3}(m_3))$ and we can get precise location information $m_1 \parallel m_2 \parallel m_3$.

3.4 Our Scheme

System initialization. According to the security parameters 1^λ chosen, AA run $Setup(1^\lambda)$ to get the system's master private key ASK and public key APK. Then, each DO goes to generate his own master private key OSK and public key OPK together with AA.

New user registration. When new users u_t join social networks, AA generates the corresponding attribute set S according to the information of u_t and executes the KeyGen (ASK, S) to generate the private key SK associated with attribute set S, then distribute SK to u_t. In addition, the AA also need to build corresponding attribute group according to the user's attribute. If the attribute sets of users u_1, u_2, u_3 are respectively {1, 2}, {1, 2, 3}, {2, 3}, the corresponding attribute groups are G(1) = {u_1, u_2}, G (2) = {u_1, u_2, u_3}, G(3) = {u_2, u_3}.

Trapdoor information release. AA goes to build a token tree according to the method referred in Sect. 2.3. Each leaf node in the tree corresponds to a user in the system and the random keys of the leaf nodes are regarded as TDKey in user's private key. According to the attribute group G(x) corresponds to $x \in A(1 \leq x \leq k)$, we can determine the minimum cover key set $MCKS_x$ and then figure out the trapdoor information $TDM_x = \{E_{RK_j}(TD_x)\}_{RK_j \in MCKS_x}$. We define RK_j as the random key, define TD_x as the attribute trapdoor of x and define E as a fast symmetric encryption algorithm, such as exclusive or operation. Release the trapdoor information TDM = $\{TDM_x\}_{x \in A}$ and token chain TCS = $\{TCS_i\}_{i \in \{1,2,\cdots,m\}}$.

Establish social contact. DOs go to distribute their master private key OSK to their own social members through a security channel (such as SSL protocol).

Release private data. We only describe how data owner DO deals with single location information m here: (1) Use three groups of randomly generated symmetric key k_1, k_2, k_3 to encrypt the three parts m_1, m_2, m_3 of location information m and we can get $E_{k_1}(m_1)$, $E_{k_2}(m_2)$, $E_{k_3}(m_3)$ (E is a symmetric encryption algorithm), then they make up the ciphertext CM of the location information. (2) Run algorithm EncryptK (APK, OPK, P_{rank}, K_{rank}) to encrypt symmetric key information K_1, K_2, K_3 according to the three kinds of attribute strategies P_1, P_2, P_3 corresponding to the ranks of intimacy, then we can get CK_1, CK_2, CK_3 and they make up the ciphertext CK of key

information. (3) Let $V = \{\rho(i) \mid 1 \leq i \leq l\}$, then calculate the trapdoor TDM_x for any attribute $x \in V$ to make up the trapdoor information $TDM_{DO} = \{TDM_x\}_{x \in V}$. (4) Release the location information file $ID_m \parallel ID_{DO} \parallel$ CM and key information file $ID_{DO} \parallel TDM_{DO} \parallel$ CK in the social network (ID_m is the unique number created for location information m and ID_{DO} is the unique number created for the data owner DO).

Data access. When user u_t wants to access location information file whose number is ID_m, SNSP goes to search for the corresponding key information file according to the number ID_{DO} and returns the location information CM, key information $TDM_{DO} \parallel$ CK and the user's token chain TCS_t. First, user u_t needs to decrypt TDM_{DO} to get the attribute trapdoor information, and then decrypt CK_{rank} in file CK with its own private key to SK and the attribute trapdoor got from the decryption of TDM_{DO}. After that, we can get the symmetric key K_{rank}. Then, we can decrypt location information file CM with symmetric key K_{rank} to get the location information. The process can be described as follow: (1) Decryption of TDM_{DO}. According to the first theorem of the token tree referred in Sect. 2.3, we can know that the key chain set KCS_t in token tree can be figured out with the trapdoor key TDKey in user u_t own private key SK and token chain set TCS_t. Obviously, if and only if the attribute set of user u_t own private key SK satisfies the attribute strategy represented by (M, ρ) can the user decrypts CK. It means that as for $W = \{\rho(i) \mid 1 \leq i \leq l \text{ and } \rho(i) \in S\}$, any attribute $x \in W$ and its corresponding attribute group G(x), there must be a user $u_t \in G(x)$. According to the third theorem of the token tree referred in Sect. 2.3, we can know that as for the minimum cover key set $MCKS_x$ of G(x), there must be a random key RK_y which could satisfy that $RK_y \in KCS_t$ and $RK_y \in MCKS_x$. It means that the user can use RK_y to decrypt TDM_x to get the corresponding attribute trapdoor TD_x of attribute x. Therefore, user u_t is able to get all the attribute trapdoors through the decryption of TDM_{DO} and then we can decrypt CK. (2) Decryption of CK. User u_t goes to run KeyUpdate (OSK, SK) to update the private key SK with owner's OSK, then run DecryptK (SK, CK) to decrypt CK to get K_{rank} with using the updated private key. (3) Decryption of CM. then, u_t runs DecryptM (K_{rank}, CM) to decrypt CM to get the corresponding location information with the K_{rank} we got.

Revocation of attributes. When it need to change user's attributes, AA goes to complete the revocation of attributes. We define the revocation set of attribute as R and the user whose attribute is revoked as u_t. The process of revocation is described as follow: (1) AA goes to update the attribute trapdoor information. As for any attribute $x \in R$, AA randomly generates new attribute trapdoor TD'_x and forms new attribute group G(x) of attribute x. Obviously, as for the user u_t whose attribute is revoked, there must be $u_t \notin G(x)$. Then, rebuild the minimum cover key sets $MCKS'_x$, generate $TDM'_x = \{E_{RK_j}(TD'_x)\}_{RK_j \in MCKS'_x}$, and take TDM'_x in the place of the original TDM_x. (2) AA goes to update APK and ASK. For any attribute $x \in R$, AA goes to update the corresponding component in the APK

$$T'_x = T_x^{TD'_x/TD_x}$$

and take TD'_x in the place of TD_x in ASK. (3) DO encrypts the ciphertext of key information again. First of all, randomly generate three groups new symmetric

encryption key k'_1, k'_2, k'_3 and form K'_1, K'_2, K'_3. Execute EncryptK () three times to get CK'_1, CK'_2, CK'_3 and form new CK' to take the place of the original CK. Then, update the information of attribute group. The process is described as follow: let the $V = \{\rho(i) \mid 1 \leq i \leq l\}$, $VR = V \cap R$ (R is the set revocation set of attribute). If $VR = \emptyset$, it doesn't need to update; If the $VR \neq \emptyset$, then for any $x \in VR$, we calculate the corresponding TDM'_x of x and take it in the place of the original TDM_x, forming new trapdoor information

$$TDM'_{DO} = \left\{TDM'_x\right\}_{x \in V}$$

(4) Release the new key information file $ID_{DO} \parallel TDM'_{DO} \parallel CK'$ again.

4 Security Analysis

4.1 Confidentiality of Attributes

The revocation of attribute in literature [12, 13] inevitably reveal the attribute information of users. In this paper, AA generates users' private key and completes the management of users' attributes independently. Therefore, if we assume AA is fully trusted, then the information of users' attributes is confidential.

4.2 Confidentiality of Data

The confidentiality of data in this paper only depends on the ciphertext's confidentiality of the location information file and the ciphertext's confidentiality of key information file. Our scheme adopts the method of mixed encryption to encrypt files, therefore, if we assume the symmetric encryption algorithm used to encrypt location information file is secure, we can get that the confidentiality of data only depends on the security of attribute encryption algorithm used to encrypt key information file and the security of revocation. Our scheme uses the WT-CP-ABE encryption algorithm, which is based on CP-ABE and makes some changes: (1) Change its previous key generation model. In our scheme, users need to get data owner DO's master private key OSK and then use OSK to update their own private key, then they can decrypt the ciptertext correctly. (2) Add the mechanism of token tree. Our paper constructs a complete binary tree and introduce the mechanism of token tree to control users' access to the attribute trapdoor TD_x so as to realize the management of attributes. Use the random key in token tree to encrypt the attribute trapdoor TD_x, therefore, if the symmetric encryption algorithm and the length of the random key adopted satisfy the requirements of security, according to the first theorem and the second theorem of the token tree referred in Sect. 2.3, we can prove the mechanism of token tree to be secure. Because the CP-ABE [10] algorithm is judgmental PBDHE mathematical problems and proved to be secure under the standard model. Therefore, WT-CP-ABE encryption algorithm is also secure under the standard model.

Our scheme uses updating the information of trapdoor to realize the revocation of user's attributes. Among them, AA randomly generate new TD_x, then use symmetric encryption to encrypt TD_x with a new random key selected from the token tree. If we assume symmetric encryption is secure, the user whose attributes are revoked cannot decrypt and get the new TD_x. Therefore, the user whose attributes are revoked cannot decrypt the data, which guarantees the security of revocation of attributes.

4.3 Resist Collusion Attack

The collusion of SNSP and the user whose attributes are revoked is one of the most common attacks [12–15]. Our scheme adopts the mechanism of token tree, DO directly encrypts the ciphertext of key information when some attributes are revoked, therefore, even if the user whose attributes are revoked in collusion with the SNSP can't he gets the updated data.

The collusion attack from unauthorized users is another threat. If two users conspire, normally they can't decrypt the ciphertext because neither of the sets of attributes they have can satisfy the conditions of decryption. But they may get some unauthorized key information if they combine their private keys. Like the methods used in literature [8, 10], our scheme embeds the random numbers in each user's private key so that the co-conspirators cannot decrypt by combining their private keys. According to the process of decryption referred in the Sect. 3.3, we can know K_1, K_2, K_3 are tied together with $e(g, g)^{\alpha s}$, and if the attacker wants to get K_1, K_2, K_3, he must go to get $e(g, g)^{\alpha s}$ first. If someone wants to get $e(g, g)^{\alpha s}$, he need to calculate $(C, \mathrm{D})/\mathrm{e}(\mathrm{g}, g)^{t\beta s}$, namely calculate $e(g, g)^{t\beta s}$. As a result, as for any attribute $\rho(\mathrm{i})$ $(\mathrm{i} \in \mathrm{I})$, the attacker must calculate

$$e(C_i, L)e(C_i', D_{\rho(i)}^{TD_{\rho(i)}})$$

Our scheme embeds the users' unique random number t in L and D_i, so it cannot complete the above calculation through the combination of different user's private key. Therefore, the co-conspirators cannot get the information of symmetric key K_1, K_2, K_3.

5 Efficiency Analysis

In this section, we will compare our scheme with EASiER in [14]. We define OSKC, OSKS OENC, ODEC to represent the time complexity for DO to generate the private key, the space complexity to store the user's private key, time complexity for DO to encrypt data and time complexity for the user to decrypt data.

When it goes to generate the private key, DO in our scheme only need to generate OSK, thus OSKC is O(1). In EASiER, however, DO need to compute each user's private key, therefore OSKC is O(na), among them, n represents the average number of social members DO have and a represents the number of attributes associated with user's private key. When it goes to the storage of private keys, users in our scheme only

need to keep their own private key SK and the OSK obtained from the DO, therefore, OSKS is O(m) + O(a), among them, m represents the average number of users who build a relationship with DO. In EASiER, however, each DO need to distribute its private key SK to users and users have to store all of them, therefore, OSKS is O(ma).

Both our scheme and EASiER adopt the method of mixed encryption, while our scheme need to use attribute encryption three times to encrypt the information file of symmetrical key. Therefore, in EASiER, OENC is O(D) + O(b) and ODEC is O (D) + O(c). OENC in our paper is O(D) + O(3b) and ODEC is O(D) + O(3c), among them, D, b, c respectively represents the size of location information file, the average number of the attributes associated with ciphertext when it encrypts and the average number of attributes needed for decryption.

From the Table 1, we can see though our scheme is greater in the time complexity of encryption and decryption process compared to the EASiER, we have obvious advantages in both the time complexity for DO to generate the private key and the space complexity to store the user's private key.

Table 1. Analysis of complexity

Scheme	*OSKC*	*OSKS*	*OENC*	*ODEC*
EASiER	$O(na)$	$O(ma)$	$O(D) + O(b)$	$O(D) + O(c)$
Our scheme	$O(1)$	$O(m) + O(a)$	$O(D) + O(3b)$	$O(D) + O(3c)$

From the Figs. 2, 3, 4 and 5, we can easily see the relationship between OSKC, OSKS, OENC, ODEC and n, m, D. When n, m is large, our scheme has more advantages in OSKC and OSKS compared with the EASiER. And with the increase in D, the increasing rate of OENC, ODEC in our scheme is the same to those in the EASiER.

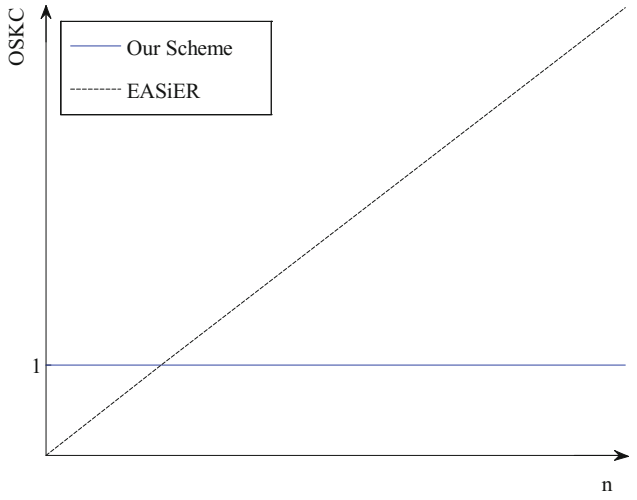

Fig. 2. Relation between OSKC and n

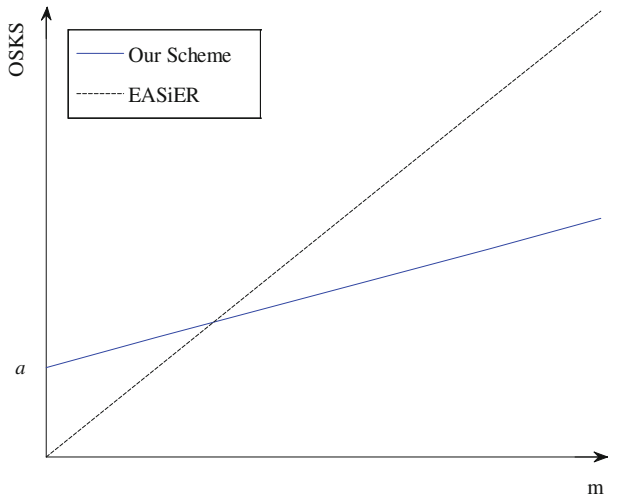

Fig. 3. Relation between OSKS and *m*

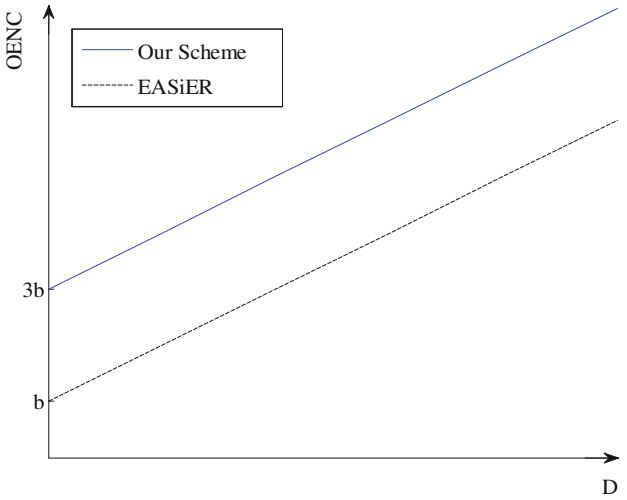

Fig. 4. Relation between OENC and *D*

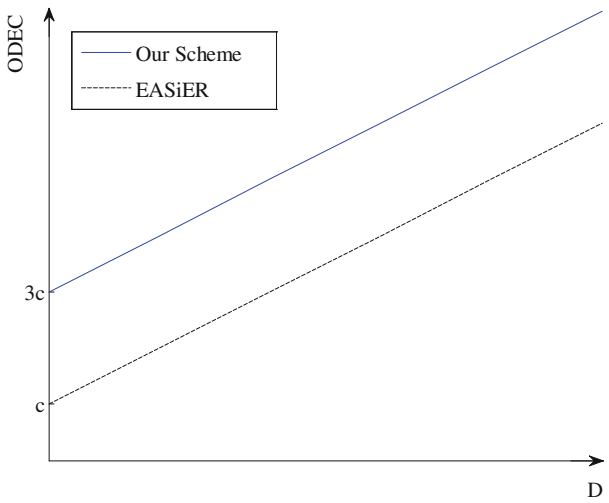

Fig. 5. Relation between ODEC and *D*

6 Conclusions

In our paper, we put forward a location privacy preserving scheme based on attribute encryption, which provides "precise", "more accurate", "fuzzy" and "private" four modes to manage the location information. Data owner DO can divide the location information into three parts according to the rank of intimacy, then use symmetric encryption and WT-CP-ABE algorithm to encrypt location information file and key information file, finally release ciphertext to social networks. By using the combination of symmetric encryption and public key encryption method, it makes the encryption of information more efficient. If and only if the key changes, the user needs to encrypt the ciphertext of key information again. However, in order to realize to decrypt the location information hierarchically, we need to use attribute encryption three times to encrypt the information of key, which increases the amount of calculation. Therefore, we will further study on how to combine symmetric encryption and attribute encryption better, reduce the tedious encryption procedures and improve the efficiency of our scheme in the future.

Acknowledgments. This work is supported by National Natural Science Foundation of China (61572521, 61272492, 61272468), Project funded by Natural Science Basic Research Plan in Shaanxi Province of China (2015JM6353) and Basic Research Plan of Engineering College of the Chinese Armed Police Force (WJY201523, WJY201613).

References

1. Lv, Z.Q., HONG, C., ZHANG, M., et al.: Privacy-perserving scheme for social networks. J. Commun. **35**(8), 23–32 (2014)
2. Chow, C.Y., Mokbel, M.F., Aref, W.G.: Casper*: query processing for location services without compromising privacy. In: Proceedings of the 32nd International Conference on Very Large Data Bases, VLDB Endowment, pp. 763–774 (2006)
3. Kido, H., Yanagisawa, Y., Satoh, T.: An anonymous communication technique using dummies for location-based services. In: Proceedings of the International Conference on Pervasive Services 2005, ICPS 2005, pp. 88–97 (2005)
4. Man, L.Y., Jensen, C.S., Huang, X., et al.: SpaceTwist: managing the trade-offs among location privacy, query performance, and query accuracy in mobile services. In: International Conference on Data Engineering, pp. 366–375 (2008)
5. Gruteser, M., Grunwald, D.: Anonymous usage of location-based services through spatial and temporal cloaking. In: International Conference on Mobile Systems, Applications, and Services, pp. 31–42 (2003)
6. Chen, W.H., Li, W.J., Zhu, J.: A model for protecting location privacy against attacks from friends in SNS. Comput. Eng. Sci. **37**(4), 692–698 (2015)
7. Boneh, D., Franklin, F.: Identity-based encryption from the Weil pairing. Adv. Cryptology **32**(3), 586–615 (2001). Crypt' 2001
8. Bethencourt, J.,Sahai, A.,Waters, B.: Ciphertext-policy attribute-based encryption. In: Proceedings of the 28th International Symposium on Security and Privacy (S&P 2007). Berkeley, CA, USA, 321–334 (2007)
9. Beimel, A.: Secure Schemes for Secret Sharing and Key Distribution. Ph.D thesis, Israel Institute of Technology, Technion, Haifa, Israel (1996)
10. Waters, B.: Ciphertext-policy attribute-based encryption: an expressive, efficient, and provably secure realization. In: Catalano, D., Fazio, N., Gennaro, R., Nicolosi, A. (eds.) PKC 2011. LNCS, vol. 6571, pp. 53–70. Springer, Heidelberg (2011). doi:10.1007/978-3-642-19379-8_4
11. Liang, X., Li, X., Lu, R., et al.: An efficient and secure user revocation scheme in mobile social networks. In: Global Telecommunications Conference (GLOBECOM 2011), pp. 1–5. IEEE (2011)
12. Hur, J., Noh, D.K.: Attribute-based access control with efficient revocation in data outsourcing systems. IEEE Trans. Parallel Distrib. Syst. **22**(7), 1214–1221 (2010)
13. Yu, S., Wang, C., Ren, K., et al.: Attribute based data sharing with attribute revocation. In: Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security, ASIACCS 2010, Beijing, China, April 13–16, 2010, pp. 261–270 (2010)
14. Jahid, S., Mittal, P., Borisov, N.: EASiER: encryption-based access control in social networks with efficient revocation. In: ACM Symposium on Information, Computer and Communications Security, ASIACCS 2011, Hong Kong, China, March 2011, pp. 411–415 (2011)
15. Zhang, M., Lv, Z., Feng, D., et al.: A secure and efficient revocation scheme for fine-grained access control in cloud storage. In: 2012 IEEE 4th International Conference on Cloud Computing Technology and Science (CloudCom), pp. 545–550. IEEE (2012)