SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

March 2020, Vol. 63 132105:1–132105:11
https://doi.org/10.1007/s11432-019-1508-6

PPLS: a privacy-preserving location-sharing scheme in mobile online social networks

Chang XU[1], Xuan XIE[1], Liehuang ZHU[1*], Kashif SHARIF[1], Chuan ZHANG[1], Xiaojiang DU[2] & Mohsen GUIZANI[3]

[1] *School of Computer Science and Technology, Beijing Institute of Technology, Beijing* 100081*, China;*
[2] *Department of Computer and Information Sciences, Temple University, Philadelphia* PA 19122*, USA;*
[3] *Department of Computer Science and Engineering, Qatar University, Doha* 2713*, Qatar*

Received 4 April 2019/Accepted 2 August 2019/Published online 10 February 2020

Abstract The recent proliferation of mobile devices has given rise to mobile online social networks (mOSNs), an emerging network paradigm that uses social networks as its main design element. As one of the most critical components in mOSNs, location sharing plays an important role in helping users share information and strengthen their social bonds, which however may compromise users' privacy, including location information and social relationship details. To address these challenges, some solutions have been proposed. However, none of them considers the privacy of inter-user threshold distance, which effectively can be used to identify users, their friends, and location information, by malicious or undesired elements of the system. To overcome this limitation, we propose a secure distance comparison protocol. Furthermore, we present a privacy-preserving location-sharing scheme (PPLS), which allows users to build more complex access control policies. The safety of our scheme is validated by the security analysis and the experimental results demonstrate the efficiency of PPLS scheme.

Keywords privacy-preservation, location-sharing, mobile online social networks

Citation Xu C, Xie X, Zhu L H, et al. PPLS: a privacy-preserving location-sharing scheme in mobile online social networks. Sci China Inf Sci, 2020, 63(3): 132105, https://doi.org/10.1007/s11432-019-1508-6

1 Introduction

The fast development & deployment of mobile computing has changed the future of communications and services, and this change has promoted different network paradigms, where mobile online social networks (mOSNs) are a leading example. Different from the traditional networks, mOSNs takes advantage of the mobility of devices and then, introduces mobile social networks as the main design ingredient. Through mOSNs, users can experience convenient communication and richer experiences [1].

The mOSNs provide various services, including location-based services (LBSs). In LBSs, because the location of a device usually represents its contextual information, geographical locations of mobile devices are exploited to provide information and entertainment services [2]. As millions of applications based on LBSs are available, users can easily obtain information such as restaurants and hotels. In fact, as a fundamental component of mOSNs, LBSs have become increasingly important and popular.

While enjoying the convenience of LBSs, the privacy threats should not be ignored. In LBSs, users are expected to update their real-time location information and share it for better services. However, disclosing the location information is dangerous, because an adversary can track an individual and infer

[*] Corresponding author (email: liehuangz@bit.edu.cn)

his/her preferences. This threat becomes more serious in mOSNs as users' locations can be correlated with their profiles [3]. Hence, it is vital to protect users' location privacy in mOSNs.

To address these problems, a series of studies have been performed. A MobiShare system was presented by Wei et al. [4] that allows users to share location information flexibly. Inspired by [4], Liu et al. [5, 6] proposed a system called N-Mobishare, which uses a location server and a social network server to manage users' location information and identity information, respectively. Li et al. [7] proposed MobiShare+ which reduces the security risk of MobiShare. In 2016, Shen et al. [8] used Bloom Filter to achieve privacy preservation and provided a system called BMobishare. In 2017, Li et al. [9] proposed a more secure location-sharing scheme, multiple location servers are used to protect users' social network privacy. In 2018, Xiao et al. [10] designed a new centralized location-sharing system without a third-party. The aforementioned systems support two kinds of queries, i.e., friends' queries and strangers' queries, and also satisfy access control policy.

However, these mechanisms are not perfect. Firstly, the threshold distance is a personal preference of each user (to establish a social circle), but this is used as public information for location service entities in the system. When the threshold distance set by a user is a special number or the threshold distance set for different targets is in a special data group, the adversary can track the data or data group to identify users. Secondly, threshold distance is used by a user to determine with whom they are willing to share the location. However, some schemes use broadcast encryption to share personal location information, which violates the distance-based access control policy. Finally, it is far from actual application requirements that all systems mentioned above use a single threshold distance for all friends. Users may wish to set different threshold distances for different friends. In our scheme, the RSA algorithm and Paillier encryption are exploited. Although some key management schemes are proposed [11–14], they do not apply to our system.

Our contributions. Motivated by these limitations, we propose a privacy-preserving location-sharing (PPLS) scheme in mOSNs. The contributions are described as follows.

(1) In some of the previously proposed mechanisms, a user can only set a single threshold distance for all friends. However, this technique does not meet the actual needs. To improve the practicability of the system, our scheme allows users to set different threshold distances for different friends. In our scheme, users can use a more flexible strategy to achieve access control.

(2) Because existing studies do not consider the privacy of the threshold distance, to get more personal information of users, an adversary can easily be used to collect threshold distances. To overcome this defect, we propose a secure distance comparison protocol to execute encrypted distance comparison and prevent location servers from determining this sensitive data.

(3) Based on the proposed secure distance comparison protocol, we propose the PPLS scheme. In PPLS, users are allowed to set different threshold distances for different friends and broadcast encryption is not used. Moreover, users can request for particular friends. Diverse queries are supported in the proposed PPLS system.

The rest of this paper is organized as follows. Section 2 provides the system models and design goals. Section 3 presents the building blocks including the proposed secure distance comparison protocol. Section 4 introduces the PPLS scheme and Section 5 gives its security analysis. Furthermore, performance analysis is provided in Section 6, and some related studies are introduced in Section 7. Finally, a conclusion is presented in Section 8.

2 System models and design goals

This section presents a formal system architecture, system workflows, and a threat model for location privacy. We also identify and list the security goals for the proposed scheme.

2.1 System architecture

Figure 1 depicts the system architecture where three main entities interact with each other.

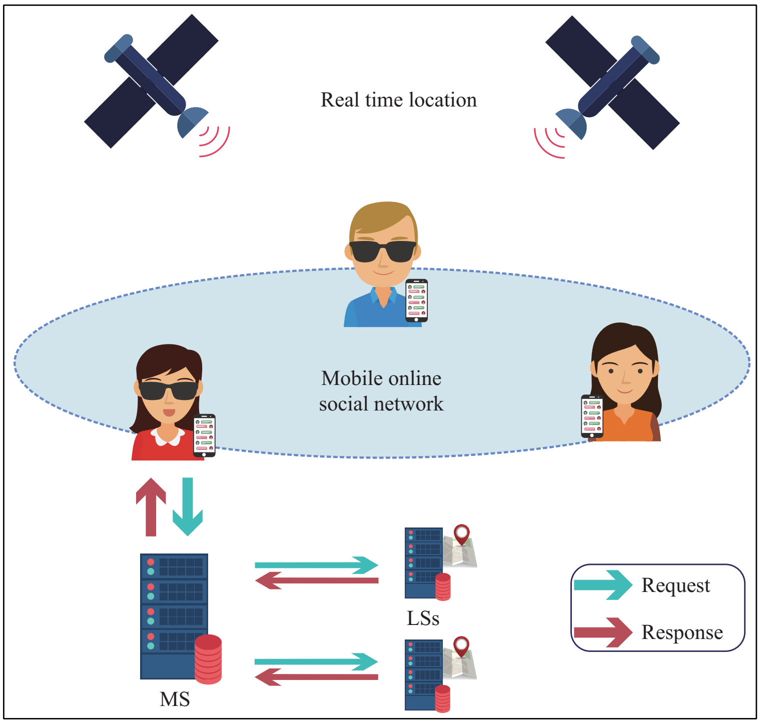

Figure 1 (Color online) System architecture.

(1) Users. The users of mOSNs with mobile devices can communicate with mobile online social network sever directly. They can get their locations from GPS and request for locations of specific friends, nearby friends, and strangers.

(2) Mobile online social network server (MS). These servers are responsible for managing user's online social relationships, such as profiles and friend lists. MS can communicate with users and location servers directly.

(3) Location servers (LSs). These servers primarily manage users' location information. They calculate location distances and related tasks of finding users within a certain area, which are assigned by MS. LSs communicate with MS directly, but different LSs are not allowed to cooperate for information exchange.

Constraints. In our system, MS should not be aware of the users' locations. Moreover, LSs are not aware of the users' identity-related information. Users may submit three types of queries: (1) request for particular friends' locations, (2) request for nearby friends' locations, and (3) request for nearby strangers' locations.

2.2 System workflows

In light of the proposed architecture, five main workflows are defined.

(1) Users must initially register with MS for location-based service. The registration process requires submitting personal identification information and making effective proof of authenticity. Moreover, users must also define their access control policies. MS maintains a database and processes a user's personal information. Using pseudo-identities and initial location information, MS also registers all users with the LS.

(2) When arriving at a new place or after a specified time, users need to update their information. In this regard, MS maintains the new relationships and threshold distances of users, whereas LSs maintain the new location information.

(3) When a user intends to obtain the location of a friend, they submit a query for that particular user. If the requester meets the access control policies of their friends, they can obtain the location information.

(4) When a user intends to obtain nearby friends current location information, they submit a query for

friends within a certain distance. If the user meets the access control policies of these required friends, they can get the desired information.

(5) In the case of a user requiring nearby stranger's current location, they submit a query for strangers within a specific distance. If the user meets the access control policy of strangers (within distance), they can get the locations of these strangers.

2.3 Threat model

Out of the listed entities (i.e., users, MS, LSs), users are considered to be dishonest. This means that they may try to access the server they do not have the permission to access and find the location of a target user. Moreover, we assume that MS and LSs are honest but curious, i.e., they will follow the scheme formally, but try to obtain as much sensitive information as possible. For example, MS may want to find the location of users and LSs may want to obtain sensitive information of users. We suppose that MS and LSs may be compromised by an adversary, but not at the same time. This means that MS and LSs will not collide with each other. The assumption is reasonable because it is extremely difficult for an adversary to control two servers at the same time.

2.4 Security goals

Using the defined threat model as a guiding principle, the security goals for location-sharing system are defined as follows:

(1) The system should protect the user's location information from MS and other unauthorized users. The users' locations cannot be leaked to friends or strangers who do not satisfy the predefined access policy.

(2) MS provides social relationships related service and should not be able to determine (directly or indirectly) the user locations.

(3) Location servers provide location-based services and should not know the users' social network information and/or identity information.

3 Building blocks

In this paper, the main challenge to solve is to implement location-based services while preserving users' privacy. In the proposed PPLS scheme, the user sets threshold distances for different friends & strangers and the threshold values may vary with different targets. It is important to note that, these values may indicate personal emotion tendency towards different targets and location service providers can collect this data to infer such personal information. Therefore, the threshold distance should be kept private in addition to the actual location. To solve this problem, we propose a secure distance comparison protocol based on Paillier encryption. The scheme also makes use of RSA encryption, which is elaborated in a nutshell for comparative understanding.

3.1 RSA encryption

RSA encryption is a widely used public-key cryptosystem for securing data transmission, where a public and private key pair is used for encryption and decryption, respectively. The process is summarized as follows.

Choose two large prime numbers p and q, compute $n = pq$. Select random integer e such that $1 < e < \lambda(n)$ and $\gcd(e, \lambda(n)) = 1$, where $\lambda(n) = (p-1)(q-1)$ and gcd is the greatest common divisor. Compute $d = e^{-1}(\bmod(\lambda(n)))$. The public key is (n, e) and the private key is (p, q, d).

Encryption. Assume that M is a message to encrypt. First, turn M (un-padded plain text) into an integer m (padded plain text) by padding scheme. The ciphertext is $c = m^e(\bmod\ n)$.

Decryption. Let c be the ciphertext to decrypt, m can be recovered by computing $c^d = (m^e)^d = m(\bmod\ n)$. Also, the plain text message M can be recovered by reversing the padding scheme.

3.2 Paillier encryption

Paillier public-key cryptosystem is a classical homomorphic semantically secure public-key cryptosystem that is used in proposed secure distance comparison protocol. This subsection outlines the basic technique of Paillier public-key cryptosystem.

Choose two large prime numbers p and q and compute $n = pq$. Select random integer g, $g \in Z^*_{n^2}$, ensure $\gcd(L(g^\lambda \bmod n^2), n) = 1$, where $L(x) = \frac{x-1}{n}$, $\lambda = \mathrm{lcm}(p-1, q-1)$, and lcm is the lowest common multiple. The public key is (n, g) and the private key is (p, q).

Encryption. Assume that m is a message to be encrypted where $0 \leqslant m \leqslant n$. Select random $r < n$, then the ciphertext is $c = g^m \cdot r^n \bmod n^2$.

Decryption. Let c be the ciphertext to decrypt, where $c \in Z^*_{n^2}$, the plain text message is $m = \frac{L(c^\lambda \bmod n^2)}{L(g^\lambda \bmod n^2)} \bmod n$.

Paillier public-key cryptosystem has the following properties.

Homomorphic addition of plain texts. We can give the value of $E(m_1 + m_2)$ through $E(m_1)$ and $E(m_2)$ without knowing m_1 and m_2.

$$D(E(m_1, r_1) \cdot E(m_2, r_2) \bmod n^2) = m_1 + m_2 \bmod n.$$

Homomorphic multiplication of plain texts. We can give the value of $E(m_1 m_2)$ through $E(m_1)$ and m_2 without knowing m_1.

$$D(E(m_1, r_1)^{m_2} \bmod n^2) = m_1 m_2 \bmod n.$$

3.3 Secure distance comparison protocol

In our system, LSs need to compare the distance between two users with the corresponding threshold distance to effectively provide services. To preserve users' privacy, we propose a secure distance comparison protocol (as shown in Protocol 1) based on [15, 16]. Let $d_{\text{threshold}}$ be the threshold distance, g be a generator of a cyclic group M, and d_{actual} be the actual distance. We set $d_{\text{threshold}}$ and d_{actual} as integers. Let G be a key generation algorithm, PE be the Paillier encryption algorithm, and PD be the Paillier decryption algorithm. R is the space of random coins, S is a probabilistic polynomial time algorithm with $S(1^k, \mathrm{PK}) \subset Z$, and k is the security parameter.

Protocol 1 Security distance comparison protocol

```
Input: Threshold distance $d_{\text{threshold}}$; Actual distance $d_{\text{actual}}$.
Output: $d_{\text{threshold}} > d_{\text{actual}}$ as TRUE or FALSE.
MS generates the key pair $(\mathrm{sk}_m, \mathrm{pk}_m) \leftarrow G(1^k)$ and a random value $r \leftarrow R$. Let $c \leftarrow \mathrm{PE}_{\mathrm{pk}_m}(d_{\text{threshold}} g; r)$. MS sends $(\mathrm{pk}_m, c)$ to LS;
LS generates random $s \leftarrow S$, $r' \leftarrow R$, computes
    $c' \leftarrow (c \cdot \mathrm{PE}_{\mathrm{pk}_m}(-(d_{\text{actual}} + i)g; 0))^s \cdot \mathrm{PE}_{\mathrm{pk}_m}(0; r')$
    $= (\mathrm{PE}_{\mathrm{pk}_m}(d_{\text{threshold}} g; r) \cdot \mathrm{PE}_{\mathrm{pk}_m}(-(d_{\text{actual}} + i)g; 0))^s \cdot \mathrm{PE}_{\mathrm{pk}_m}(0; r')$
    $= \mathrm{PE}_{\mathrm{pk}_m}(s(d_{\text{threshold}} - (d_{\text{actual}} + i))g; r^s \circ r')$
  for $i = 1, 2, \ldots, n-1$, LS computes $c'$ and sends $c'$ to MS;
MS outputs $d_{\text{threshold}} > d_{\text{actual}}$ as TRUE, if and only if $\mathrm{PD}_{\mathrm{pk}_m}(c') = 0$ is found. Otherwise output FALSE.
```

4 Privacy-preserving location-sharing (PPLS) scheme

To preserve the user's location and social network privacy, the scheme utilizes encryption keys generated by different system entities. The details of each step are given below and Table 1 lists the notations used in them.

Initialization. Each user has their identifier ID and a public-private key pair $(\mathrm{pk}_u, \mathrm{sk}_u)$ which can later be updated. Assume users' group is represented as $U = \{u_1, u_2, \ldots, u_z\}$. MS stores a social network

Table 1 Summary of notations

Symbol	Description
ID	A user's social network identifier
PID	A user's pseudo-identifier
MS	Mobile online social network server
LSs	Location servers
df	Threshold distance for a friend
ds	Threshold distance for strangers
$(\mathrm{pk}_u, \mathrm{sk}_u)$	A user's public-private key pair
$(\mathrm{pk}_m, \mathrm{sk}_m)$	MS's public-private key pair
$(\mathrm{pk}_s, \mathrm{sk}_s)$	LS's public-private key pair
tl	The time length for LS to save a record
ts	Time stamp
t	Users' location update cycle
(x, y)	Location of a user
$\mathrm{dis}(u_i, u_j)$	Distance between u_i and u_j
PE	Paillier encryption algorithm
PD	Paillier decryption algorithm

graph $G = (V, E)$, where V is a set of vertices which represents users and E is a set of edges which represents the relationships between users. LS has a public-private key pair $(\mathrm{pk}_s, \mathrm{sk}_s)$ and all users know LS's pk_s.

Registration. When a user u_i with an identifier ID intends to use the system's services, they need to register with the MS first. Registration is in the form of $(\mathrm{ID}, C_{\mathrm{pk}_s}(x_i, y_i), C_{\mathrm{pk}_s}(\mathrm{pk}_u), \mathrm{Flist}, (\mathrm{df}_{i,1}, \mathrm{df}_{i,2}, \ldots, \mathrm{ds}), \mathrm{ts}, \mathrm{Sig}(\mathrm{ID}, \mathrm{ts}))$, where $C_{\mathrm{pk}_s}(x_i, y_i)$ and $C_{\mathrm{pk}_s}(\mathrm{pk}_u)$ are u_i's location & public key (respectively) encrypted by LS's public key, Flist is u_i's friend list, $\mathrm{df}_{i,1}$ is u_i's threshold distance for friend u_1 within which they are willing to share location with u_1, ds is the threshold distance for strangers with which u_i is willing to reveal its location to strangers, ts is a time stamp, and $\mathrm{Sig}(\mathrm{ID}, \mathrm{ts})$ is a signature generated on ts. MS holds a database to save users' threshold distances.

MS confirms the request. If the signature is valid, MS generates a registration request to LS. The request is in the form of $\left(\mathrm{PID}, C_{\mathrm{pk}_s}(x, y), C_{\mathrm{pk}_s}(\mathrm{pk}_u), \mathrm{tl}\right)$, in which PID is u_i's pseudo-identity generated by $\mathrm{AES}(\mathrm{ID}, \mathrm{rt})$ and rt is a random value. tl is the time limit for which the record will be held. LSs can timely remove the expired data and reduce storage overhead. The value of tl should be set slightly larger than the update cycle.

Update. For each time period t, users need to update their information. Similar to the registration content, each user sends a message to MS in the form of $(\mathrm{ID}, C_{\mathrm{pk}_s}(x, y), C_{\mathrm{pk}_s}(\mathrm{pk}_u), \mathrm{Flist}, (\mathrm{df}_{i,1}, \mathrm{df}_{i,2}, \ldots, \mathrm{ds}), \mathrm{ts}, \mathrm{Sig}(\mathrm{ID}, \mathrm{ts}))$, where $C_{\mathrm{pk}_s}(x, y)$, Flist, and $(\mathrm{df}_{i,1}, \mathrm{df}_{i,2}, \ldots, \mathrm{ds})$ represent user's new locations encrypted by LS's public key, new friendship, and new threshold distances, respectively. Without updating pk_u, the adversary can associate users' PIDs by tracing pk_u. If the signature is valid, MS sends $(\mathrm{PID}, C_{\mathrm{pk}_s}(x, y), C_{\mathrm{pk}_s}(\mathrm{pk}_u), \mathrm{tl})$ to LSs. LSs save related information in their database.

Request for particular friends. If a user u_i with an identifier ID wants to obtain the location(s) of their friend(s) $(f_1, f_2, \ldots, f_M)$, u_i submits a query for friends' locations in the form of $(\mathrm{ID}, C_{\mathrm{pk}_s}(x_i, y_i), \mathrm{pf}, (f_1, f_2, \ldots, f_M))$ to MS, where pf represents the request type. To handle this request, MS first recovers the pseudo-identity $P_{\mathrm{ID}} = (\mathrm{PID}_1, \mathrm{PID}_2, \ldots, \mathrm{PID}_M)$ corresponding to $(f_1, f_2, \ldots, f_M)$. Then, MS randomly divides P_{ID} into Q subsets $P_{\mathrm{ID}}^1, P_{\mathrm{ID}}^2, \ldots, P_{\mathrm{ID}}^Q$ with different sizes, satisfying $P_{\mathrm{ID}} = P_{\mathrm{ID}}^1 \cup P_{\mathrm{ID}}^2 \cup \cdots \cup P_{\mathrm{ID}}^Q$, to prevent the adversary from knowing u_i's friend relationships. For $P_{\mathrm{ID}}^j = (\mathrm{PID}_1, \mathrm{PID}_2, \ldots, \mathrm{PID}_N)$, MS computes $(c_{1,i}, c_{2,i}, \ldots, c_{N,i}) = (\mathrm{PE}_{\mathrm{pk}_m}(\mathrm{df}_{1,i}g; r_1), \mathrm{PE}_{\mathrm{pk}_m}(\mathrm{df}_{2,i}g; r_2), \ldots, \mathrm{PE}_{\mathrm{pk}_m}(\mathrm{df}_{N,i}g, r_N))$, and sends $(\mathrm{PID}, C_{\mathrm{pk}_s}(x_i, y_i), \mathrm{pf}, P_{\mathrm{ID}}^j, (c_{1,i}, c_{2,i}, \ldots, c_{N,i}), \mathrm{pk}_m)$ to LS_j, where LS_j is the jth location server in LSs. After receiving the request, LS_j performs the following steps:

(1) Decrypt $C_{\mathrm{pk}_s}(x_i, y_i)$ to get u_i's current location (x_i, y_i).

(2) Calculate the distances between u_i and his/her friends and save as $(d_1, d_2, \ldots, d_N) = (\mathrm{dis}(u_i, \mathrm{PID}_1),$

$\mathrm{dis}(u_i, \mathrm{PID}_2), \dots, \mathrm{dis}(u_i, \mathrm{PID}_N))$.

(3) Choose parameters s and r'. For $c_{1,i}$, calculate $c'_{1,i} = ((c_{1,i} \cdot \mathrm{PE}_{\mathrm{pk}_m}(-(d_1+p)\,g;0))^s \cdot \mathrm{PE}_{\mathrm{pk}_m}(0;r'))$. Let $p = 1, 2, \dots, n-1$ and send corresponding $c'_{1,i}$ to MS.

If and only if there exists p such that $\mathrm{PD}_{\mathrm{sk}_m}(c'_{1,i}) = 0$, then $d_1 < \mathrm{df}_{1,i}$ and u_i satisfies PID_1's access control policy, otherwise u_i does not satisfy the policy. MS finds all u_i's friends for whom u_i satisfies their access control policies. Then LS_j sends those friends' encrypted locations to MS. After collecting all results returned by LSs, MS sends u_i the ciphertexts. u_i decrypts the ciphertexts and gets their requested friend's location.

Request for friends within specific distance. If a user u_i with identifier ID wants to find friends' locations within a certain distance, then a query for friends' locations is submitted in the form of $(\mathrm{ID}, C_{\mathrm{pk}_s}(x_i, y_i), f, l)$ to MS, where f indicates the type of request. Similar to request for particular friends' locations, after grouping friends randomly, MS sends $(\mathrm{PID}, C_{\mathrm{pk}_s}(x_i, y_i), f, P^j_{\mathrm{ID}}, (c_{1,i}, c_{2,i}, \dots, c_{N,i}), \mathrm{pk}_m, l)$ to LS_j. When receiving the request, LS_j performs the following steps:

(1) Decrypt $C_{\mathrm{pk}_s}(x_i, y_i)$ to get u_i's current location (x_i, y_i).

(2) Calculate the distances between u_i and all of their friends, and save as $(d_1, d_2, \dots, d_N) = (\mathrm{dis}(u_i, \mathrm{PID}_1), \mathrm{dis}(u_i, \mathrm{PID}_2), \dots, \mathrm{dis}(u_i, \mathrm{PID}_N))$.

(3) Choose parameters s and r'. For $c_{1,i}$, calculate $c'_{1,i} = ((c_{1,i} \cdot \mathrm{PE}_{\mathrm{pk}_m}(-(d_1+p)\,g;0))^s \cdot \mathrm{PE}_{\mathrm{pk}_m}(0;r'))$. Let $p = 1, 2, \dots, n-1$ and send corresponding $c'_{1,i}$ to MS.

If and only if there exists p such that $\mathrm{PD}_{\mathrm{sk}_m}(c'_{1,i}) = 0$, then $d_1 < \mathrm{df}_{1,i}$ and u_i satisfies PID_1's access control policy. Furthermore, if $d_1 < l$, then f_1's location will be returned. MS finds all these friends and gets their encrypted locations from LS_j. After collecting the results returned by all LSs, MS sends the final response to u_i, which decrypts the ciphertext with their own private key sk_u and gets the friends' locations.

Request for strangers within specific distance. If a user u_i wants to find location of stranger(s) who are within l distance from them, then u_i submits a strangers' locations query $(\mathrm{ID}, C_{\mathrm{pk}_s}(x_i, y_i), s, l)$ to MS. Here, s is the request type. Because there are too many unfamiliar users around u_i, MS sends LSs a query (PID, all, l) first. LSs find all users within l distance away from u_i and feedback the result. Then, MS eliminates u_i's friends and strangers randomly and sends $(\mathrm{PID}, C_{\mathrm{pk}_s}(x_i, y_i), s, P^j_{\mathrm{ID}}, (c_{1,i}, c_{2,i}, \dots, c_{N,i}), \mathrm{pk}_m)$ to LS_j. Assuming a stranger u_2 is within l distance away from u_i. u_2's location is (x_2, y_2) and u_2's threshold distance for strangers is ds_2. If and only if $\mathrm{dis}(u_i, u_2) < \mathrm{ds}_2$, LS_j returns u_2's encrypted location to MS. MS sends the final result to u_i.

5 Security analysis

The security analysis is provided based on the threat model and security goals. In PPLS, we assume that MS and LSs. Hence, they do not collude with each other and are not compromised by the adversary at the same time.

Access control. PPLS allows users to set different threshold distances for different targets. Because MS and LSs are assumed to be honest but curious, they will follow the protocol formally. That means, only the users who satisfy the access policy can receive the location information and identity information of friends/strangers.

Identity privacy. In PPLS, LSs should not have any knowledge of users' identity-related information. Pseudo-identity is used when users send update messages or queries. Thus, anonymity is achieved. Although threshold distances may leak identity information (indirectly) of users to the adversary, homomorphic encryption is used to encrypt the sensitive data. Thus, users' identity privacy is well-preserved.

Location privacy. MS may collude with dishonest users and attempt to obtain the location information of a particular user illegally. When receiving the registration/update messages from users or receiving the responses from LSs, MS has the chances to access users' locations. PPLS encrypts users' locations using asymmetric encryption, which protects location information from MS.

Social network privacy. The privacy of the social network is preserved by two approaches, which are described as follows.

(1) When a user requests for particular friends or friends/strangers within specific distances, MS will divide the friends/strangers into random subsets and send these sets to different LSs. These subsets have different sizes and will be sent randomly to LSs. Furthermore, dummy users can be added to the original set. As a result, each LS can only get part of the friend list with dummy users. Because we assume that LS will not collude with each other, LSs are prevented from knowing users' social networks.

(2) For each time t, users need to update their information. During this phase, MS assigns each user a new pseudo-identifier, which is different from the original one. As a result, after the time t, for different queries from the same user, the user's pseudo-identifier and his/her friends' pseudo-identifiers become different. Therefore, it is impossible for LSs to determine the information on users' social networks.

6 Experimental evaluation

The proposed PPLS scheme uses several of encryption and decryption steps. To evaluate real-time performance, we conducted several experiments.

6.1 Implementation

In our system, three cryptography schemes are implemented: digital signature, asymmetric encryption, and homomorphic encryption. We use RSA [17] with 1024-bit key size for data encryption, RSA PKCS1-v1-5 for signature, and Paillier with 1024-bit key size for homomorphic encryption. Our simulation is implemented on an Intel Xeon E3-1230v3 running at 3.4 GHz with 8 GB 2133 GHz memory. We used Python 3.5.0 to implement the proposed algorithms. Some PyPI packages are used in our cryptography schemes: pycrypto for signature, asymmetric encryption and phe for Paillier encryption.

In our experiments, mobile devices use many effective techniques to obtain locations, such as GPS or cellular geo-location. We assume that the threshold distance can be set as $10, 20, \ldots, 100$ m with steps of 10 m or $100, 200, \ldots, 1000$ m with steps of 100 m. For friends, users may consider choosing a smaller value as the threshold distance. For strangers, users may choose a larger value as the threshold distance. We assume that the actual distances between users are within 1000 m and are generated randomly.

6.2 Evaluation

As the RSA signing technology used in the registration and update phases can be replaced by any other signing algorithms, we do not analyze the registration and update phase.

The response time of the system to request for particular friends is related to the number of friends the user requests. The response time to request for friends or strangers within a specific distance is related to the size of the request area and the user density within the scope. In essence, this parameter is also based on several users requested. Therefore, we observe the time spent for the entire request process and the time spent for a secure distance comparison protocol against a different number of requested users. We conduct each experiment 10 times and calculate the average values. The results are shown in Figures 2 and 3, respectively.

It can be observed from the results that the time spent on the request process approximately increases linearly with the number of users requested, about 0.75 s per 10 individuals. When the number of the requested users increases to 150, the entire request process costs 10.63 s. The secure distance comparison protocol execution time also approximately increases linearly with the number of users requested, about every 10 individuals with 0.7 s. When the number of the requested users increases to 150, the secure distance comparison protocol process costs 9.86 s. We can see the time spent in implementing the secure distance comparison protocol takes up a large percentage of the system's time (to generate a response). The protocol time-consumption is mainly focused on determining the size relationship between the actual distance and the threshold distance, the traversal encryption of the actual distance in LS and the response decryption in MS.

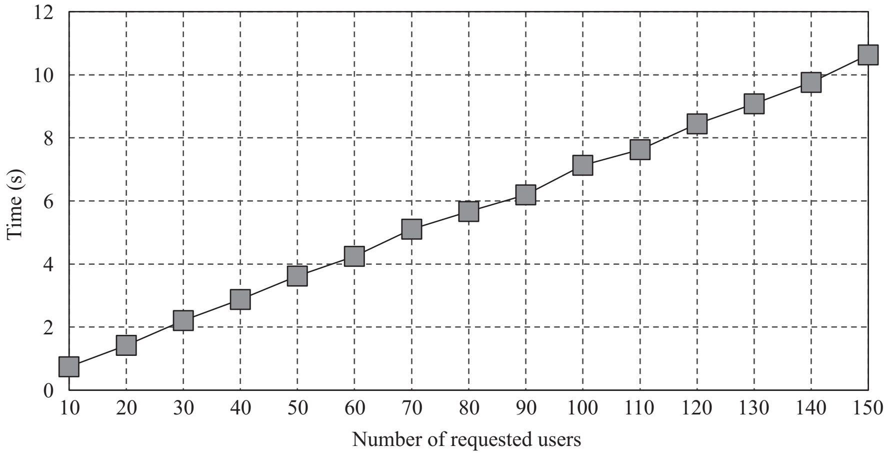

Figure 2 Entire request process.

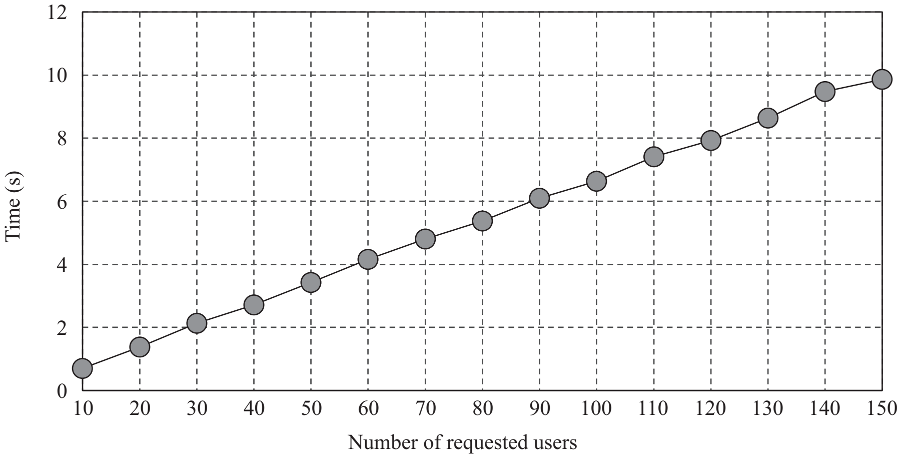

Figure 3 Secure distance comparison protocol process.

6.3 Comparison

In this subsection, we compare Mobishare [4], N-Mobishare [5,6], Location-sharing systems with enhanced privacy (LSEP) [9] with our proposed system PPLS.

(1) In Mobishare, cellular towers act as trusted centers and complete cryptographic operations. Other system architectures are simpler because cellular towers are not included in the system models.

(2) The performance on the mobile device side: N-Mobishare and LESP need to use broadcast encryption to achieve information sharing, while Mobishare needs to perform symmetric encryptions n-times. In our system, users only encrypt their locations and sign their messages sent to MS in initialization and registration phases.

(3) The performance on the social network server-side: In our system, the social network server implements the secure distance comparison protocol, which takes up a large percentage of system execution time. In Mobishare and N-Mobishare, the social network server needs to store some location information, while in LESP the social network server has to encrypt the sensitive data.

(4) The performance on the location server side. In our system, each location server executes the secure distance comparison protocol when they are required to compare the vehicles' true distances and the threshold distances, but the comparison tasks are distributed to multiple servers.

7 Related work

In recent years, privacy preservation in social networks has garnered great attentions [18]. With the mobile devices widely been used, mOSNs have experienced an explosive development. Because a user's location is an important information used in mOSNs, the issue of protecting users' location privacy has also received considerable attention. Until now, many studies on location privacy protection [19, 20] have been done, such as location anonymity and information hiding [21]. Location anonymity is an effective technique for location privacy protection and there are two types of methods to achieve it. (1) K-anonymity: The fundamental premise is to mix the real user's location information into $k-1$ other anonymous users' location information, which confuses the adversary. This approach is proposed in [22] by Sweeney in 2002 and then, Gruteser et al. [23] used it for location privacy protection. Kido et al. [24] extended K-anonymity and introduced the concept of virtual location. (2) Location encryption: The main idea of location encryption is to encrypt the users' location information with some encryption algorithms, such as the algorithm proposed by Khoshgozaran et al. [25] using Hilbert curves to encrypt the original location.

By combining the aforementioned methods, a series of studies have been proposed. In 2007, Smoke-Screen [26] proposed a scheme to protect users' location privacy and provide location-sharing services for users in mOSNs. Subsequently, Wei et al. [4] proposed MobiShare, which supports users sharing location information flexibly. In MobiShare, social network server and location server store users' profiles and location information separately. Hence, neither of the two severs know the complete information of the users. However, this scheme cannot protect users' social network topologies. Later, based on MobiShare, several mechanisms are proposed, such as N-MobiShare [5, 6], MobiShare+ [7], and B-MobiShare [8]. In N-MobiShare, the cellular tower is not treated as a core component of the system. Social network server takes cellular tower's task and forwarded users' requests to the location server. N-MobiShare uses broadcast encryption to share offline keys to users' friends. Although N-Mobishare has a simpler structure than MobiShare, it do not solve the problem which MobiShare encountered. That is, the location server can still get users' social network topologies in the query phase. Inspired by Wei et al.'s solution, Li et al. [7] found that in MobiShare the pseudo-identity of the querying user can be known by LSs in the friend's query. Hence, they proposed an improved mechanism named MobiShare+ [7]. Besides dummy locations and identities, this mechanism employes dummy queries. It applies a private set intersection protocol to prevent individual information leaked between the social network server (SNS) and location-based server. MobiShare+ overcomes the defect of MobiShare and N-MobiShare. However, it incurres excessively long processing time. To solve this problem and improve the transmission efficiency, Shen et al. [8] proposed B-MobiShare. Bloom Filter is used in this scheme to replace the private set intersection protocol in MobiShare+ and the time cost is reduced. However, B-MobiShare is less efficient than expected, the time cost is still high. In 2017, Li et al. [9] proposed a system with enhanced privacy in mOSNs, using multiple location servers to prevent insider attack launched by the service providers. In 2018, Xiao et al. [10] designed a new centralized location-sharing system, using location-storing social network server instead of employing the third-party server. However, all the above mechanisms do not treat the threshold distance as sensitive data and work with a single threshold distance for users to set for all of their friends, which is unrealistic in real social networks.

8 Conclusion

As privacy preservation of location sharing in mOSNs is an important issue, in this paper, we propose PPLS that is used for protecting users' location privacy from MS and preserving users' social network privacy from LSs. The scheme allows users to set different threshold distances for different friends and to enjoy a more flexible access control policy. To implement this access control policy, a secure distance comparing protocol is presented. New queries are designed for particular friends to permit users sharing locations with friends. The security analysis shows that PPLS is secure under a comprehensive security model. Moreover, the experimental evaluation demonstrates that the time complexity increases linearly with an increase in users. In our scheme, the response time of the system still needs to be further improved. We will design more efficient comparison protocols in the future research. Besides, with the rapid development of edge computing, in new scenarios, how to achieve location sharing has become a hot research topic. In future, we will try to provide location-sharing services while protecting users' privacy under new application scenarios.

Acknowledgements This work was supported by National Natural Science Foundation of China (Grant Nos. 61972037, 61402037, U1836212, 61872041) and Graduate Technological Innovation Project of Beijing Institute of Technology (Grant No. 2019CX10014).

References

1 Chen J, Wu G, Shen L, et al. Differentiated security levels for personal identifiable information in identity management system. Expert Syst Appl, 2011, 38: 14156–14162

2 Dinh H T, Lee C, Niyato D, et al. A survey of mobile cloud computing: architecture, applications, and approaches. Wirel Commun Mob Comput, 2013, 13: 1587–1611

3 Barkhuus L, Dey A K. Location-based services for mobile telephony: a study of users' privacy concerns. In: Proceedings of International Conference on Human-Computer Interaction, Zurich, 2003. 702–712

4 Wei W, Xu F Y, Li Q. Mobishare: flexible privacy-preserving location sharing in mobile online social networks. In: Proceedings of IEEE INFOCOM, 2012. 2616–2620

5 Liu Z L, Li J, Chen X F, et al. New privacy-preserving location sharing system for mobile online social networks. In: Proceedings of the 8th International Conference on P2P, Parallel, Grid, Cloud and Internet Computing, Compiegne, 2013. 214–218

6 Liu Z L, Luo D J, Li J, et al. N-Mobishare: new privacy-preserving location-sharing system for mobile online social networks. Int J Comput Math, 2016, 93: 384–400

7 Li J W, Li J, Chen X F, et al. {MobiShare}+: security improved system for location sharing in mobile online social networks. J Internet Serv Inf Secur, 2014, 4: 25–36

8 Shen N, Yang J, Yuan K, et al. An efficient and privacy-preserving location sharing mechanism. Comput Standards Interfaces, 2016, 44: 102–109

9 Li J, Yan H Y, Liu Z L, et al. Location-sharing systems with enhanced privacy in mobile online social networks. IEEE Syst J, 2017, 11: 439–448

10 Xiao X, Chen C, Sangaiah A K, et al. CenLocShare: a centralized privacy-preserving location-sharing system for mobile online social networks. Future Generation Comput Syst, 2018, 86: 863–872

11 Xiao Y, Rayi V K, Sun B, et al. A survey of key management schemes in wireless sensor networks. Comput Commun, 2007, 30: 2314–2341

12 Du X J, Xiao Y, Guizani M, et al. An effective key management scheme for heterogeneous sensor networks. Ad Hoc Netw, 2007, 5: 24–34

13 Du X J, Guizani M, Xiao Y, et al. Transactions papers a routing-driven elliptic curve cryptography based key management scheme for heterogeneous sensor networks. IEEE Trans Wirel Commun, 2009, 8: 1223–1229

14 Kandah F, Zhang W Y, Du X J, et al. A secure key management scheme in wireless mesh networks. In: Proceedings of IEEE International Conference on Communications, Kyoto, 2011. 1–5

15 Lipmaa H. Verifiable homomorphic oblivious transfer and private equality test. In: Proceedings of International Conference on the Theory and Application of Cryptology and Information Security. Berlin: Springer, 2003. 416–433

16 Golle P. A private stable matching algorithm. In: Proceedings of International Conference on Financial Cryptography and Data Security. Berlin: Springer, 2006. 65–80

17 Bellare M, Rogaway P. The exact security of digital signatures - how to sign with RSA and rabin. In: Proceedings of International Conference on the Theory and Application of Cryptographic Techniques, Saragossa, 1996. 399–416

18 Yan C L, Ni Z Y, Cao B, et al. UMBRELLA: user demand privacy preserving framework based on association rules and differential privacy in social networks. Sci China Inf Sci, 2018, 62: 039106

19 Ju X, Shin K G. Location privacy protection for smartphone users using quadtree entropy maps. J Inf Priv Secur, 2015, 11: 62–79

20 Rao U P, Girme H. A novel framework for privacy preserving in location based services. In: Proceedings of International Conference on Advanced Computing & Communication Technologies (ACCT), 2015. 272–277

21 Das A K. A secure and effective biometric-based user authentication scheme for wireless sensor networks using smart card and fuzzy extractor. Int J Commun Syst, 2017, 30: e2933

22 Sweeney L. k-anonymity: a model for protecting privacy. Int J Unc Fuzz Knowl Based Syst, 2002, 10: 557–570

23 Gruteser M, Grunwald D. Anonymous usage of location-based services through spatial and temporal cloaking. In: Proceedings of the 1st International Conference on Mobile Systems, Applications, and Services, San Francisco, 2003. 31–42

24 Kido H, Yanagisawa Y, Satoh T. Protection of location privacy using dummies for location-based services. In: Proceedings of the 21st International Conference on Data Engineering Workshops, Tokyo, 2005. 1248

25 Khoshgozaran A, Shahabi C. Blind evaluation of nearest neighbor queries using space transformation to preserve location privacy. In: Proceedings of the 10th International Symposium on Advances in Spatial and Temporal Databases, Boston, 2007. 239–257

26 Cox L P, Dalton A, Marupadi V. Smokescreen: flexible privacy controls for presence-sharing. In: Proceedings of the 5th International Conference on Mobile Systems, Applications, and Services (MobiSys 2007), San Juan, 2007. 233–245