EURASIP Journal on Wireless Communications and Networking

RESEARCH Open Access

Location-sharing protocol for privacy protection in mobile online social networks

Ou Ruan, Lixiao Zhang and Yuanyuan Zhang*

*Correspondence: circle0519@hotmail.com
School of Computers, Hubei University of Technology, No. 28, Nanli Road, Hong-shan District, 430068 Wuhan, China

Abstract

Location-based services are becoming more and more popular in mobile online social networks (mOSNs) for smart cities, but users' privacy also has aroused widespread concern, such as locations, friend sets and other private information. At present, many protocols have been proposed, but these protocols are inefficient and ignore some security risks. In the paper, we present a new location-sharing protocol, which solves two issues by using symmetric/asymmetric encryption properly. We adopt the following methods to reduce the communication and computation costs: only setting up one location server; connecting social network server and location server directly instead of through cellular towers; avoiding broadcast encryption. We introduce dummy identities to protect users' identity privacy, and prevent location server from inferring users' activity tracks by updating dummy identities in time. The details of security and performance analysis with related protocols show that our protocol enjoys two advantages: (1) it's more efficient than related protocols, which greatly reduces the computation and communication costs; (2) it satisfies all security goals; however, most previous protocols only meet some security goals.

Keywords: Location privacy, Location sharing, Mobile online social networks, Smart cities

1 Introduction

Smart city [1–3] is an agglomeration of urban society that employs intelligent technologies to improve people's life, work and communication. A key aspect of smart cities is mobile online social networks (mOSNs) [4–8] that are popular mainstream platforms for information and content sharing among people. Location sharing is an important component of mOSNs and also has attracted much attention recently. No matter where a person is, nearby people can be matched by applications such as WeChat, Facebook and Twitter [9], which change traditional social ways and help people broaden their circle of friends [10].

With the increasing popularity of location-based services, the risk of leakage of users' privacy also increases. According to users' location information, users' activity track can be obtained by criminals. Furthermore, users' private information such as health condition, home address, work unit can be inferred [11]. This threat becomes even more serious when location information is combined with social relationship

[12, 13]. Users may hesitate to share their locations over mOSNs if their privacy is not protected [14]. Therefore, protecting users' privacy is our first priority in mOSNs. Many studies have been proposed and they are mainly divided into the following two categories:

(1) *K-anonymity* The typical methods in references [15] and [16] are used to obscure the real location by generating $(k-1)$ virtual locations. That is to say, k positions are generated, including one real position and $(k-1)$ dummy positions to prevent attackers from identifying the real position. However, firstly, some protocols set up cellular towers, which are used to connect the social network server S_{OSN} and the location server S_{LS}, and will increase communication costs of the whole system. Secondly, in some protocols, query results contain the real identity, which leaks the identity privacy. Thirdly, some protocols can't verify the identity of sender, that is to say, these protocols can't conduct identity authentication.

(2) *Dummy identity* The methods in references [17, 18] are to only share location but hide identity. The method anonymizes users' identities by adopting pseudonyms. However, some protocols set up multiple location servers and adopt broadcast encryption, which increases communication and computation costs. On the other hand, in some protocols, location servers can infer users' activity tracks and further learn sensitive information such as health state.

From the above analysis, these methods suffer two constraints: (1) they are inefficient and have high communication and computation costs. (2) they ignore some security goals. In the paper, we present a new protocol, which solves two issues by using symmetric/asymmetric encryption properly. Firstly, in our protocol, only one location server is set up, broadcast encryption isn't needed, and S_{OSN} and S_{LS} are connected directly instead of connecting through cellular towers, thus we greatly reduce computation and communication costs. Secondly, inquirers only get dummy identities rather than real identities, which protects users' identity privacy. Thirdly, we prevent location server from inferring users' activity track by updating dummy identities in time. Finally, we conduct identity authentication to prevent impersonation attacks.

Compared with related articles, our advantages are as follows:

- Our protocol greatly reduces computation and communication costs. In our protocol, firstly, only one location server rather than multiply location servers is set up to reduce communication and computation costs. Secondly, we apply symmetric encryption instead of broadcast encryption to reduce communication and computation costs. Thirdly, cellular towers aren't set up to reduce communication costs.
- Our protocol satisfies all security goals; however, most previous protocols only meet some security goals.

This paper is organized as follows. The related works are described in Sect. 2. In Sect. 3, some preliminaries are provided. In Sect. 4, our new method is proposed. Result and discussion are presented in Sect. 5. Conclusion is described in Sect. 6.

2 Related works

According to the underlying cryptographic techniques, there are two common methods: *K-anonymity*, *Dummy identity*.

K-anonymity The typical method was first proposed by Sweeney et al. [15] in 2002, and then, Gruteser et al. [16] used it for location privacy protection. Kido et al. [17] extended *K-anonymity* and introduced the concept of virtual location. But the method *K-anonymity* has an obvious disadvantage: it incurs great communication and computation costs.

In 2004, a protocol [19] was proposed, which can hide users' positions in sensitive areas by location updation. But users may be traced by location servers. An improved model [20] called *CacheCloak* solves the problem. *CacheCloak* can make location data anonymous in real time. The activity track can be predicted by a trusted server from previous location information and be submitted to S_{LS}. However, one user's predicted activity track intersects that of others, thus preventing one user's activity track from being captured by untrusted location servers.

In 2015, *BMobishare* model [21] was proposed by Shen et al., which employs Bloom Filter to mask sensitive data. It employs Bloom Filter, thus a malicious user cannot obtain unauthorized privacy information. But it ignores identity privacy and identity authentication. Identity privacy: query results contain the real identities which leak the identity privacy. Identity authentication: if a server doesn't verify received information, an attacker may send a location message to the server by pretending the identity of a legitimate user. In 2019, Chen et al. presented a new model [22] to protect identity privacy and conduct identity authentication.

Dummy identity The typical method *Dummy identity* was first proposed by Cox et al. [18] in 2007, called *SmokeScreen* model, where a user ID_i sets up his access control policies df_{ID_i} for friends and ds_{ID_i} for strangers. Friends or strangers can obtain users' locations if they satisfy the access control policies. However, in *SmokeScreen* model, there is only one server used to store personal information such as social network relation and location, if a malicious user colludes with the server, the identity information and the corresponding location information will be obtained by the malicious user. In 2012, Wei et al. [23] proposed *Mobishare* model to solve this problem. The model sets two servers: S_{LS} and S_{OSN}. S_{LS} is used to store location data and S_{OSN} is used to store personal information such as social relation. That is to say, neither S_{OSN} nor S_{LS} has a complete information including the users' identities and locations. So users' privacy is protected even if location server or social network server colludes with malicious users. But users' social relations can be inferred by location server. In 2013, Li et al. [24] proposed *Mobishare+* model, which can be seen as an improved mechanism based on Mobishare model. It employs dummy queries and private set intersection protocol to prevent S_{OSN} and S_{LS} from learning individual information. In 2013, Liu et al. [25] proposed *N-Mobishare* model based on *Mobishare* model. Compared with *Mobishare* model and *Mobishare+* model, cellular towers aren't set up in *N-Mobishare* that is used to connect S_{OSN} and S_{LS}, thus *N-Mobishare* reduces communication costs. In 2014, an improved model [26] of *N-Mobishare* was proposed, which can prevent location server from inferring social relation of users. S_{OSN} generates a set containing dummy identities of an inquirer's all friends and adds some randomly dummy identities to further anonymize the inquirer's real social relation. But the model adopts broadcast encryption which requires user to dynamically change his sharing decryption key when a friend is added or revoked, which will lead to great communication and computation costs.

In 2017, a new model called UDPLS [27] was proposed to prevent S_{OSN} from learning location information and S_{LS} from learning about users' social relationships. Li et al. [28] proposed a model with multiple location servers. When a user queries locations of friends, the social relation of the user will be randomly divided into multiple sets by S_{OSN}, and these sets will be sent to different location servers. However, in the location update stage, the location is encrypted by the secret keys of every S_{LS} and sent to every S_{LS}, thus it results in great communication and computation costs. In addition, although S_{LS} only stores users' anonymous identities, their activity tracks also can be inferred. In 2020, Xu et al. [29] presented a model with multiple location servers to protect the activity track privacy. However, due to the use of multiple location servers, this protocol has higher communication and computation costs.

3 Preliminaries

3.1 System model

Figure 1 shows the system architecture, which contains three entities.

Users A user is an entity who locates the current location through his/her mobile phone. He/She can share his/her current location with the nearby friends or strangers, and can also query the locations of friends or strangers in his/her self-defined range.

Online Social Network Server S_{OSN}. It provides online social services and manages every user's personal materials such as his friends set, friend's access control policy df_{ID_i} and stranger's access control policy ds_{ID_i}. In S_{OSN}, each user has a corresponding dummy identity.

Location Server S_{LS}. It manages users' dummy identities and locations and returns the location information to the queriers.

3.2 Threat model

Users may be dishonest and try to get all the location information which meet their needs, but some of the location information are beyond users' right.

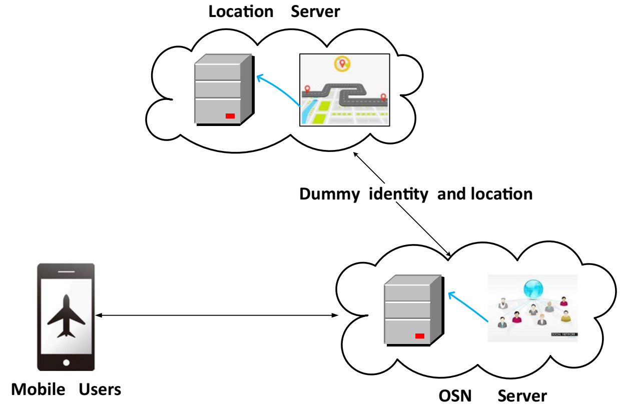

Fig. 1 System architecture in mobile online social networks (mOSNs)

Table 1 Table of notations

Notation	Description
ID_i	User i's social network identifier
$FID_i \leftarrow_R \{0,1\}^k$	A random bit-string FID_i selected from a set of k bit-string
S_{LS}	Location server
S_{OSN}	Social network server
(x_i,y_i)	User ID_i's real location
df_{ID_i}	User ID_i's friend-case threshold distance
ds_{ID_i}	User ID_i's stranger-case threshold distance
G	A social network graph stored in S_{OSN}
(pk_{ID_i}, sk_{ID_i})	User ID_i's public key and secret key pair
(pk_{osn}, sk_{osn})	The public key and secret key pair of S_{OSN}
(pk_{LS}, sk_{LS})	Location server's public key and secret key pair
dist((x_1,y_1),(x_2,y_2))	The function calculates the distance between (x_1,y_1) and (x_2,y_2)
min(x,y)	The function that computes the minimum of x and y
$s_{OSN} = Sig_{sk_{osn}}(FID_i, ts)$	A signature s_{OSN} generated with the user's secret key sk_{osn} over (FID_i,ts)
$Ver_{pk_{osn}}(s_{OSN})$	Verify the correctness of the signature s_{OSN} with the user's public key pk_{osn}

S_{OSN} and S_{LS} are both considered as "honest-but-curious." S_{OSN} may attempt to get the location information which should be managed by S_{LS}, and S_{LS} may try to gain the social relation which is stored in S_{OSN}.

In our security model, we assume that the users don't collude with S_{OSN} and S_{LS}. On the other hand, we assume that S_{OSN} and S_{LS} do not collude and can't obtain the information of each other.

3.3 Security goals

In our model, the security goals have the following six aspects:

- *Authorized access* A user's location can't be accessed by friends or strangers who do not conform to the user's access control policies df_{ID_i} or ds_{ID_i}.
- *Identity privacy* Users can only get friends' or strangers' dummy identities as query results, and they can't collude with S_{OSN} to get the real identities of friends or strangers.
- *Social relation privacy* S_{LS} should be prevented from obtaining users' personal materials such as the social relation.
- Location privacy S_{OSN} should be prevented from obtaining users' locations.
- *Activity track privacy* S_{LS} should be prevented from inferring users' activity tracks.
- *Identity authentication* Some measures should be taken to prevent users' identities from being impersonated.

3.4 Notation

We present the main notations in Table 1.

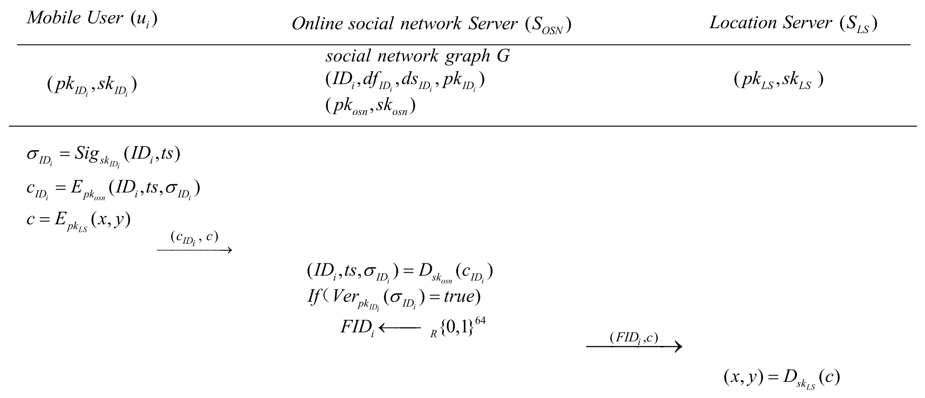
Mobile User (u_i)
Online social network Server (S_{OSN})
Location Server (S_{LS})
(pk_{ID_i}, sk_{ID_i})
social network graph G
$(ID_i, df_{ID_i}, ds_{ID_i}, pk_{ID_i})$
(pk_{osn}, sk_{osn})
(pk_{LS}, sk_{LS})
$\sigma_{ID_i} = Sig_{sk_{ID_i}}(ID_i, ts)$
$c_{ID_i} = E_{pk_{osn}}(ID_i, ts, \sigma_{ID_i})$
$c = E_{pk_{LS}}(x, y)$
$\xrightarrow{(c_{ID_i},\, c)}$
$(ID_i, ts, \sigma_{ID_i}) = D_{sk_{osn}}(c_{ID_i})$
$If(\,Ver_{pk_{ID_i}}(\sigma_{ID_i}) = true)$
$FID_i \longleftarrow_R \{0,1\}^{64}$
$\xrightarrow{(FID_i, c)}$
$(x, y) = D_{sk_{LS}}(c)$

Fig. 2 User registration

4 Our methods

4.1 A new location sharing protocol for privacy protection

The details of our location-sharing protocol are as follows:

(a) *System initialization*

(1) A user ID_i generates his public key and secret key pair (pk_{ID_i}, sk_{ID_i}), defines his access control policy df_{ID_i} and ds_{ID_i}. In the friends' location query phase, df_{ID_i} refers to the condition that other users must satisfy if they want to access user ID_i's location. Similarly, ds_{ID_i} is applied to the strangers' location query stage. Personal data $(ID_i, pk_{ID_i}, df_{ID_i}, ds_{ID_i})$ is stored at S_{OSN}.

(2) S_{OSN} generates his public key and secret key pair (pk_{osn}, sk_{osn}), stores a social network graph G, which involves all users' social relation.

(3) S_{LS} generates his public key and secret key pair (pk_{LS}, sk_{LS}).

(b) *User registration* The details of registration are described in Fig. 2.

(1) User ID_i generates a signature with his secret key $\sigma_{ID_i} = Sig_{sk_{ID_i}}(ID_i, ts)$, where ts is a timestamp, and encrypts authentication information $(ID_i, ts, \sigma_{ID_i})$ and his current location *(x,y)* to generate $C_{ID_i} = E_{pk_{osn}}(ID_i, ts, \sigma_{ID_i})$, $c = E_{pk_{LS}}(x, y)$, then sends (C_{ID_i}, c) to S_{OSN}.

(2) Upon receiving (C_{ID_i}, c), S_{OSN} decrypts C_{ID_i}, gets $(ID_i, ts, \sigma_{ID_i})$ and verifies whether the digital signature σ_{ID_i} is correct. If the authentication passes, S_{OSN} randomly generates a dummy identity FID_i and stores the record $(ID_i, FID_i, df_{ID_i}, ds_{ID_i}, pk_{ID_i})$ at S_{OSN}, then sends (FID_i, c) to S_{LS}.

(3) S_{LS} decrypts c to acquire location information *(x,y)* and stores the record $(FID_i, (x, y))$ at S_{LS}.

(c) *Location updation* The details of location updation are described in Fig. 3.

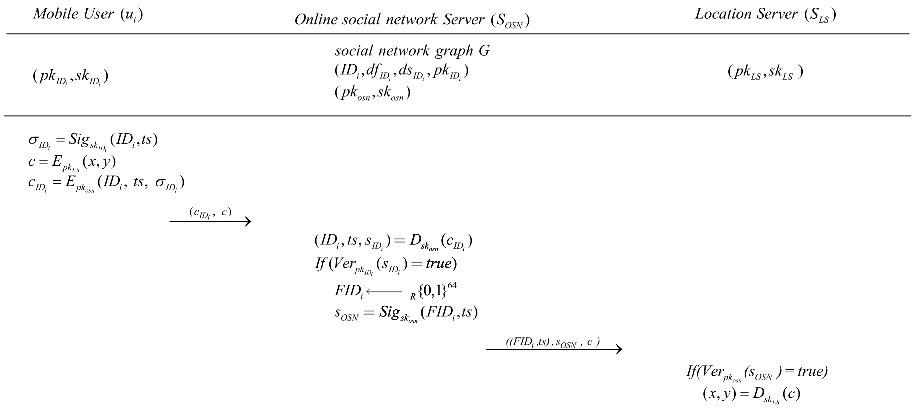
Mobile User (u_i)
Online social network Server (S_{OSN})
Location Server (S_{LS})
(pk_{ID_i}, sk_{ID_i})
social network graph G
$(ID_i, df_{ID_i}, ds_{ID_i}, pk_{ID_i})$
(pk_{osn}, sk_{osn})
(pk_{LS}, sk_{LS})
$\sigma_{ID_i} = Sig_{sk_{ID_i}}(ID_i, ts)$
$c = E_{pk_{LS}}(x, y)$
$c_{ID_i} = E_{pk_{osn}}(ID_i,\ ts,\ \sigma_{ID_i})$
$(c_{ID_i},\ c)$
$(ID_i, ts, s_{ID_i}) = D_{sk_{osn}}(c_{ID_i})$
$If\,(Ver_{pk_{ID_i}}(s_{ID_i}) = true)$
$FID_i \longleftarrow {}_R\{0,1\}^{64}$
$s_{OSN} = Sig_{sk_{osn}}(FID_i, ts)$
$((FID_i, ts), s_{OSN},\ c)$
$If(Ver_{pk_{osn}}(s_{OSN}) = true)$
$(x, y) = D_{sk_{LS}}(c)$

Fig. 3 Location updation

(1) User ID_i generates the authentication information $C_{ID_i} = E_{pk_{osn}}(ID_i, ts, \sigma_{ID_i})$, encrypts the current newest location $(x,\ y)$, $c = E_{pk_{LS}}(x, y)$ and sends them to S_{OSN}.

(2) Upon receiving the information, S_{OSN} decrypts C_{ID_i} and verifies whether the digital signature σ_{ID_i} is correct. If the authentication passes, S_{OSN} randomly generates a dummy identity FID_i and replaces previous FID_i. That is to say, each user has a unique dummy identity. Then the latest dummy identity is updated in the personal information $(ID_i, FID_i, df_{ID_i}, ds_{ID_i}, pk_{ID_i})$ at S_{OSN}. S_{OSN} signs the latest dummy identity as well as the timestamp ts, it can be expressed as $\sigma_{OSN} = Sig_{sk_{osn}}(FID_i, ts)$. S_{OSN} sends $((FID_i, ts), \sigma_{OSN}, c)$ to S_{LS}.

(3) Upon receiving $((FID_i, ts), \sigma_{OSN}, c)$, S_{LS} verifies whether σ_{OSN} is correct. If the verification passes, the latest FID_i and (x,y) are stored at S_{LS}.

(d) *Friends' location query* The details of friends' location query are described in Fig. 4. At this stage, the inquirers can query the locations of friends according to their own needs. $'l'$ can be understood as the distance threshold formulated by an inquirer.

(1) Inquirer ID_i generates a one-time symmetric key k and encrypts k and his own current location information $(x,\ y)$ to generate c, where $c = E_{pk_{LS}}(x, y, k)$. k is a bit string consisting of 128 bit random zeros or ones and can be expressed as $k \leftarrow_R \{0, 1\}^{128}$. The inquirer also encrypts the query condition $(ID_i, 'F', l)$ to generate $C_{ID_i} = E_{pk_{osn}}(ID_i, 'F', l)$, where $'F'$ stands for friend query. The inquirer sends (C_{ID_i}, c) to S_{OSN}.

(2) Upon receiving the information, S_{OSN} decrypts C_{ID_i} and gets $(ID_i, 'F', l)$. S_{OSN} matches inquirer's friends set and matches each friend's (ID_j, FID_j, df_{ID_j}). Then S_{OSN} calculates each friend's dm_{ID_j}, which is expressed as $dm_{ID_j} = min(df_{ID_j}, l)$, and matches (FID_j, dm_{ID_j}) of all friends to form the set S. Suppose the number of the inquirer's friends is m, $S = ((FID_1, dm_{ID_1}), .., (FID_m, dm_{ID_m}))$. Then S_{OSN} encrypts the set S to generate $C_{LS} = E_{pk_{LS}}(S)$ and sends (C_{LS}, c) to S_{LS}.

(3) Upon receiving the information, S_{LS} decrypts c and C_{LS} and matches the corresponding position (x_j, y_j) and checks which member can meet the

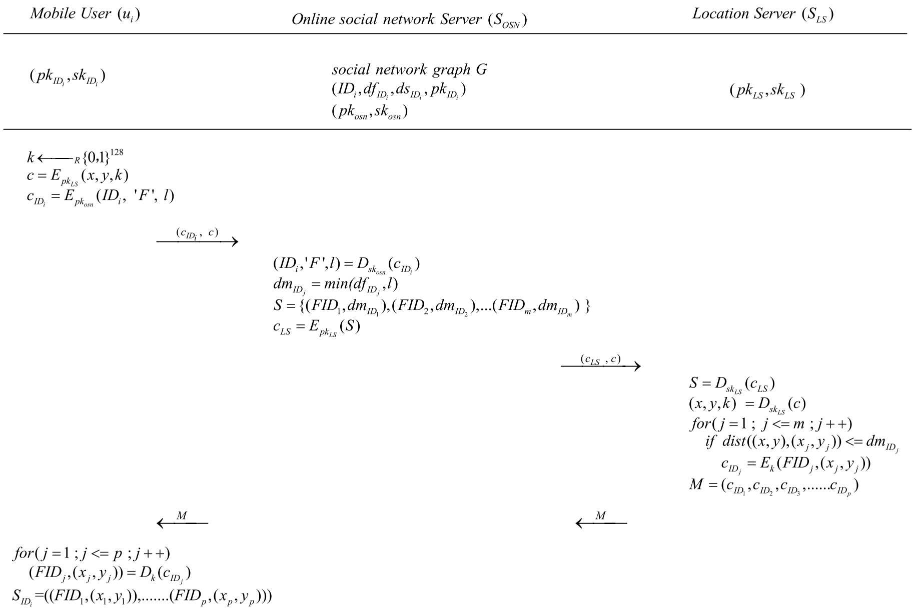
Mobile User (u_i)
Online social network Server (S_{OSN})
Location Server (S_{LS})
(pk_{ID_i}, sk_{ID_i})
social network graph G
$(ID_i, df_{ID_i}, ds_{ID_i}, pk_{ID_i})$
(pk_{osn}, sk_{osn})
(pk_{LS}, sk_{LS})
$k \longleftarrow_R \{0,1\}^{128}$
$c = E_{pk_{LS}}(x, y, k)$
$c_{ID_i} = E_{pk_{osn}}(ID_i, 'F', l)$
(c_{ID_i}, c)
$(ID_i, 'F', l) = D_{sk_{osn}}(c_{ID_i})$
$dm_{ID_j} = min(df_{ID_j}, l)$
$S = \{(FID_1, dm_{ID_1}), (FID_2, dm_{ID_2}), ...(FID_m, dm_{ID_m})\}$
$c_{LS} = E_{pk_{LS}}(S)$
(c_{LS}, c)
$S = D_{sk_{LS}}(c_{LS})$
$(x, y, k) = D_{sk_{LS}}(c)$
$for(j = 1; j <= m; j++)$
$if\ dist((x, y), (x_j, y_j)) <= dm_{ID_j}$
$c_{ID_j} = E_k(FID_j, (x_j, y_j))$
$M = (c_{ID_1}, c_{ID_2}, c_{ID_3}, ...c_{ID_p})$
M
M
$for(j = 1; j <= p; j++)$
$(FID_j, (x_j, y_j)) = D_k(c_{ID_j})$
$S_{ID_i} = ((FID_1, (x_1, y_1)), ...(FID_p, (x_p, y_p)))$

Fig. 4 Friends' location query

condition $dist((x, y), (x_j, y_j)) <= dm_{ID_j}$. If a member FID_j meets the condition, S_{LS} encrypts the location information $c_{ID_j} = E_k(FID_j, (x_j.y_j))$. Suppose there are p qualified members, which are represented as a set M, $M = (c_{ID_1}, c_{ID_2}, c_{ID_3}, ..., c_{ID_p})$. S_{LS} sends the set M to S_{OSN} and S_{OSN} forwards it to the inquirer.

(4) The inquirer decrypts and gets all of the location information $(FID_j, (x_j, y_j))$ in the set M.

(e) *Strangers' location query* The details of strangers' location query are described in Fig 5. At this stage, the inquirers can query the location of strangers according to their own needs. $'l'$ can be understood as the distance threshold formulated by an inquirer.

(1) Inquirer ID_i generates a one-time symmetric key k and encrypts k and his own current location information (x, y) to generate c, where $c = E_{pk_{LS}}(x, y, k)$. k is a bit string consisting of 128 bit random zeros or ones and can be expressed as $k \leftarrow_R \{0, 1\}^{128}$. The inquirer also encrypts the query condition $(ID_i,' S', l)$ to generate $C_{ID_i} = E_{pk_{osn}}(ID_i,' S', l)$, $'S'$ stands for stranger query. The inquirer sends (C_{ID_i}, c) to S_{OSN}.

(2) Upon receiving the information, S_{OSN} decrypts C_{ID_i} and gets $(ID_i,' S', l)$. S_{OSN} matches inquirer's strangers set and matches each stranger's (ID_j, FID_j, ds_{ID_j}). Then S_{OSN} calculates each stranger's dm_{ID_j} which is expressed as $dm_{ID_j} = min(ds_{ID_j}, l)$, and matches (FID_j, dm_{ID_j}) of all strangers form the set S. Suppose the number of the inquirer's strangers is m, then the set S can be expressed as $S = ((FID_1, dm_{ID_1})..(FID_m, dm_{ID_m}))$. Then S_{OSN} encrypts the set S to generate $C_{LS} = E_{pk_{LS}}(S)$ and sends (C_{LS}, c) to S_{LS}.

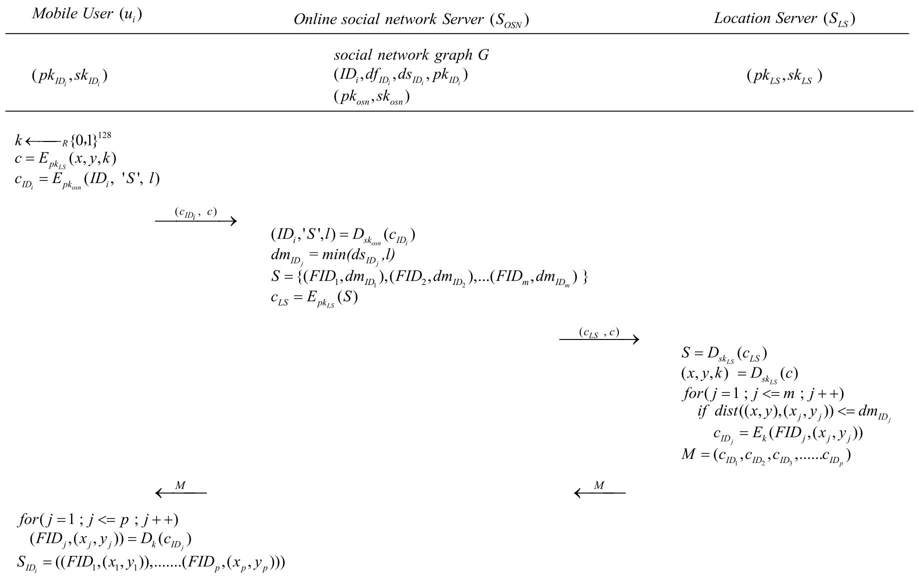
Mobile User (u_i)
Online social network Server (S_{OSN})
Location Server (S_{LS})
(pk_{ID_i}, sk_{ID_i})
social network graph G
$(ID_i, df_{ID_i}, ds_{ID_i}, pk_{ID_i})$
(pk_{osn}, sk_{osn})
(pk_{LS}, sk_{LS})
$k \xleftarrow{}_R \{0,1\}^{128}$
$c = E_{pk_{LS}}(x, y, k)$
$c_{ID_i} = E_{pk_{osn}}(ID_i, 'S', l)$
(c_{ID_i}, c)
$(ID_i, 'S', l) = D_{sk_{osn}}(c_{ID_i})$
$dm_{ID_j} = min(ds_{ID_j}, l)$
$S = \{(FID_1, dm_{ID_1}), (FID_2, dm_{ID_2}), \ldots (FID_m, dm_{ID_m})\}$
$c_{LS} = E_{pk_{LS}}(S)$
(c_{LS}, c)
$S = D_{sk_{LS}}(c_{LS})$
$(x, y, k) = D_{sk_{LS}}(c)$
$for(j = 1; j <= m; j++)$
$if\ dist((x, y), (x_j, y_j)) <= dm_{ID_j}$
$c_{ID_j} = E_k(FID_j, (x_j, y_j))$
$M = (c_{ID_1}, c_{ID_2}, c_{ID_3}, \ldots\ldots c_{ID_p})$
M
M
$for(j = 1; j <= p; j++)$
$(FID_j, (x_j, y_j)) = D_k(c_{ID_j})$
$S_{ID_i} = ((FID_1, (x_1, y_1)), \ldots\ldots (FID_p, (x_p, y_p)))$

Fig. 5 Strangers' location query

(3) Upon receiving the information, S_{LS} decrypts c, C_{LS} and matches the corresponding position (x_j, y_j), and checks which member can meet the condition $dist((x, y), (x_j, y_j)) <= dm_{ID_j}$. If a member FID_j meets the condition, S_{LS} encrypts the location $C_{ID_j} = E_k(FID_j, (x_j.y_j))$. Suppose there are p qualified members that are represented as a set M, $M = (c_{ID_1}, c_{ID_2}, c_{ID_3}, ..., c_{ID_p})$. S_{LS} sends the set M to S_{OSN} and S_{OSN} sends M to the inquirer.

(4) The inquirer decrypts and gets all of the location information $(FID_j, (x_j, y_j))$ in the set M.

4.2 Security analysis

In this section, we give the security analysis as follows:

(1) *Authorized access* Every user defines his access conditions df_{ID_i} which is used to friends' location query and ds_{ID_i} which is used to strangers' location query. Only when an inquirer satisfies a user's df_{ID_i} or ds_{ID_i}, he can obtain the user's location information. Otherwise, the user's location information will be protected and can't be learned by the inquirer.

(2) *Identity privacy* When users query friends' or strangers' locations, S_{OSN} matches and sends all friends' dummy identities rather than real identities to the S_{LS}, then S_{LS} sends them to inquirers. Thus, in the process of query, inquirers can't get friends' or strangers' real identities, so that identity privacy is protected.

(3) *Social relation privacy* If a user is constantly updating his or her locations, *FID* is constantly changed. Then when different inquirers inquiry a common friend's location, they are matched different dummy identities in S_{LS}. S_{LS} will assume that dif-

ferent users are being queried, thus records in S_{LS} are not linked to the common friend, which prevents the user's friends relation from being acquired by S_{LS}.

(4) *Location privacy* In the location updation stage and location query stage, location information is sent to S_{OSN} in the form of ciphertext, which means that the location of users are avoided to be obtained by S_{OSN}.

(5) *Activity track privacy* When a user updates his location, S_{OSN} assigns different *FID* and S_{LS} stores the newest *FID* and location. In other word, S_{LS} stores several records which belong to the same user, but S_{LS} believes every record belongs to different users due to the different *FID* and doesn't connect the records with the same user and can't infer users' activity tracks.

(6) *Identity authentication* When S_{OSN} receives a user's location, it verifies the user's identity. So the user's identity will not be impersonated.

5 Results and discussion

5.1 Implementation

We run our experiments for mobile users in a Xiaomi smartphone with Android operation system. S_{LS} is simulated with the Intel(R) Core(TM)i7-8750H 2.20-GHz CPU, and S_{OSN} is simulated with Alibaba Cloud in Ubuntu18.04 with linux 4.4.0.59. Our system is implemented using the Bouncycastle library for the cryptographic operations. The following cryptography tools are included in our implementation: SM2 encrypt/decrypt algorithm, SM2 signature/verify algorithm, SM3 hash algorithm and AES encrypt/decrypt algorithm. And we use Gaode map API to get the real geographical location. We give the running times of two main stages in our system in Table 2.

5.2 Discussion

A detailed analysis with other related protocols is given in Tables 3 and 4. We evaluate the performance in terms of two aspects: security goals, communication and computation costs.

(1) *Security goals* From Table 3, we can see that our protocol achieves all security goals, but other protocols only meet part of the following security goals.

Identity privacy users can only get friends' or strangers' dummy identities as query results, and they can't collude with S_{OSN} to get the real identities of friends or strangers. In [21, 30], enquirer can get the real identities of friends or strangers. So, they don't satisfy the security goal.

Table 2 The running times (ms) of two main stages

Stage	Entity		
	User	Social network server	Location server
Location updation	453	130	34
Location query	150	75	20

Table 3 Comparison of performances with other related protocols

Protocol	Cellular towers	Broadcast encryption	The number of S_{LS}	Authorized access	Identity privacy	Social relation privacy	Location privacy	Activity track privacy	Identity authentication
Nan Shen [21]	Yes	No	1	Yes	No	Yes	Yes	Yes	No
Jin Li [28]	No	Yes	n	Yes	Yes	Yes	Yes	No	Yes
Xi Xiao [30]	Yes	No	1	Yes	No	Yes	Yes	Yes	No
Juan Chen [22]	No	No	1	Yes	Yes	Yes	Yes	Yes	Yes
Chang Xu [29]	No	No	n	Yes	Yes	Yes	Yes	Yes	Yes
Our protocol	No	No	1	Yes	Yes	Yes	Yes	Yes	Yes

Table 4 Detailed computation and communication costs

Protocol	Communication costs		Computation costs	
	Location updation	Location query	Location updation	Location query
Nan Shen [21]	$(k+1)\lambda$	$(3p+3)\lambda$	k.Enc-asym + k.Dec-asym	p.Enc-sym + Enc-asym + p.Dec-sym + Dec-asym
Jin Li [28]	$(n+n^2)\lambda+\mu$	$(n+n^2+2p)\lambda$	n.Enc-asym + n.Dec-asym + Sig + Ver	$(n+p)$.Enc-asym + (n + p). Dec-asym
Xi Xiao [30]	$(k+1)\lambda$	$(2p+1)\lambda$	k.Enc-asym + k.Dec-asym	$(p+1)$.Enc-asym + $(p+1)$.Dec-asym
Juan Chen [22]	$(k+1)\lambda+2\mu$	$2p\lambda+4\mu$	k.Enc-asym + k.Dec-asym + Sig + Ver	p.Enc-asym + p.Dec-asym + Sig + Ver
Chang Xu [29]	$(2n^2+2n)\lambda+\mu$	$(n^2+t+2p+n)\lambda$	2n.Enc-asym + 2n.Dec-asym +Sig + Ver	$(t+n+p)$Enc-asym+$(t+p+n)$ Dec-asym
Our protocol	$3\lambda+\mu$	$(2p+4)\lambda$	2.Enc-asym + 2.Dec-asym + 2.Sig + 2.Ver	3.Enc-asym + p. Enc-sym + 3.Dec-asym + p.Dec-sym

λ: the length of elements calculated by elliptic curve encryption operation

μ: the length of elements calculated by elliptic curve signature operation

p: In the location query stage of friends and strangers, the number of members meeting the requirements of the inquirer

n: the number of location services

Enc-sym: a symmetric encryption operation

Enc-asym: an asymmetric encryption operation

Dec-sym: a symmetric decryption operation

Dec-asym: an asymmetric decryption operation

Sig: a elliptic curve signature operation

Ver: an operation to verify the validity of an elliptic curve signature

k: one real location and $(k-1)$ virtual locations generated by k anonymous

t: the number of friends assigned to each location server

Activity track privacy S_{LS} should be prevented from inferring users' activity tracks. In [28], location server can infer the activity track and further learn sensitive information such as interest and health state, which will pose a great security threat to users. In our protocol, we protect activity track privacy by updating user's dummy identities.

Identity authentication Some measures should be taken to prevent users' identities from being impersonated. In the communication process of [21, 30] which evade the identity authentication, an attacker may send a location message to servers by pretending a legitimate user's identity. If the servers do not verify the received information, they will store the wrong location message. In our protocol, social network service will conduct identity authentication to avoid this security problem.

(2) *Communication and computation costs*

Firstly, from Table 3, our protocol and [22] only need one location server and get rid of cellular towers and broadcast encryption. [21, 30] have *cellular towers* that are used to connect S_{OSN} and S_{LS} and thus increase the communication costs. [28, 29] set up *multiple location servers*. In the location updation stage and location query stage, every location information needs to be encrypted by every location server's public key, thus one location information will be encrypted multiple times by users, which will cause large communication and computation costs. *Broadcast encryption* [28] leads to great communication and computation costs. Since the ciphertext is encrypted with the key shared with the current authorized users, if a user joins or exits, in order to ensure the security of the broadcast, the keys of current authorized users must be updated, which leads to great communication and computation costs.

Secondly, Table 4 shows the detailed computation and communication costs of two main stages: the location updation stage and location query stage. *Communication costs*: (1) in the location updation stage, communication costs in our protocol are much less than other protocols. (2) in the location query stage, communication costs in our protocol are similar to [22, 30], and are much less than other protocols. *Computation costs*: (1) in the location updation stage, computation costs in our protocol are much less than other protocols because we use less asymmetric encryption than other protocols. (2) in the location query stage, computation costs in our protocol is similar to [21], and is much less than other protocols, where many asymmetric encryptions are used.

6 Conclusion

We presented an efficient privacy-preserving location sharing protocol in mOSNs for smart cities, which not only supported location sharing among friends and strangers, but also protected users' privacy. Our protocol had higher efficiency than other related protocols and achieved all security goals. In our protocol, there is a storage pressure because a new record will be generated on the location server once a user updates his location. In the future, we can further improve our protocol by finding appropriate methods to regularly delete invalid records in the location server during the location updation phase.

Abbreviations

S_{OSN}:: Social network server; S_{LS}:: Location server.

Acknowledgements
Not applicable.

Authors' contributions
The authors have contributed jointly to all parts on the preparation of this manuscript, and all authors read and approved the final manuscript.

Funding
This work is supported by the National Natural Science Foundation of China under Grants 61701173, 61672010 and 61702168, the fund of Hubei Key Laboratory of Transportation Internet of Things (WHUTIOT-2017B001), and the Ph.D. research startup foundation of Hubei University of Technology (BSQD2015028).

Availability of data and materials
Not applicable.

Declarations

Competing interests
The authors declare that they have no competing interests.

Received: 5 January 2021 Accepted: 26 April 2021
Published online: 17 May 2021

References

1. Y. Chen, X. Zou, K. Li, X. Yang, C. Chen, Multiple local 3D CNNs for region-based prediction in smart cities. Inf. Sci. **542**, 476–491 (2021). https://doi.org/10.1016/j.ins.2020.06.026
2. G. Premsankar, B. Ghaddar, M. Slabicki, M.D. Francesco, Optimal configuration of LoRa networks in smart cities, in *IEEE Transactions on Industrial Informatics* (2020), pp. 7243–7254. https://doi.org/10.1109/TII.2020.2967123
3. R.M. Sandoval, A. Garcia-Sanchez, J. Garcia-Haro, Performance optimization of LoRa nodes for the future smart city/industry. J. Wirel. Commun. Netw. **2019**(1), 200 (2019). https://doi.org/10.1186/s13638-019-1522-1
4. J. Liu, L. Fu, X. Wang, F. Tang, G. Chen, Joint recommendations in multilayer mobile social networks, in *IEEE Transactions on Mobile Computing* (2020), pp. 2358–2373. https://doi.org/10.1109/TMC.2019.2923665
5. A. M. Vegni, C. Souza, V. Loscrí, E. Hernández-Orallo, P. Manzoni, Data transmissions using hub nodes in vehicular social networks, in *IEEE Transactions on Mobile Computing* (2020), pp. 1570–1585. https://doi.org/10.1109/TMC.2019.2928803
6. Z. Sun, H. Kou, W. Huang, Privacy-aware friend finding in social network based on thumbs-up data. J Wirel. Commun. Netw. **2019**(1), 211 (2019). https://doi.org/10.1186/s13638-019-1538-6
7. C. Yan, Z. Ni, B. Cao, R. Lu, S. Wu, Q. Zhang, Umbrella: user demand privacy preserving framework based on association rules and differential privacy in social networks. Sci. China Inf. Sci. **62**(3), 39106 (2018). https://doi.org/10.1007/s11432-018-9483-x
8. X. Ju, K.G. Shin, Location privacy protection for smartphone users using quadtree entropy maps. J. Inf. Priv. Secur. **11**(2), 62–79 (2015). https://doi.org/10.1080/15536548.2015.1045372
9. H.T. Dinh, C. Lee, D. Niyato, P. Wang, A survey of mobile cloud computing: architecture, applications, and approaches. Wirel. Commun. Mob. Comput. **13**(18), 1587–1611 (2013). https://doi.org/10.1002/wcm.1203
10. N. Li, G. Chen, Sharing location in online social networks. IEEE Netw. **24**(5), 20–25 (2010). https://doi.org/10.1109/MNET.2010.5578914
11. Y. Sun, M. Chen, L. Hu, Y. Qian, M.M. Hassan, ASA: against statistical attacks for privacy-aware users in location based service. Future Gener. Comput. Syst. **70**, 48–58 (2017). https://doi.org/10.1016/j.future.2016.06.017
12. L. Barkhuus, B. Brown, M. Bell, S. Sherwood, M. Hall, M. Chalmers, From awareness to repartee: sharing location within social groups, in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (2008), pp. 497–506. https://doi.org/10.1145/1357054.1357134
13. E. Toch, J. Cranshaw, P.H. Drielsma, J.Y. Tsai, P.G. Kelley, J. Springfield, L. Cranor, J. Hong, N. Sadeh, Empirical models of privacy in location sharing, in *Proceedings of the 12th ACM International Conference on Ubiquitous Computing* (2010), pp. 129–138. https://doi.org/10.1145/1864349.1864364
14. L. Barkhuus, A.K. Dey, Location-based services for mobile telephony: a study of users' privacy concerns, in *Human-Computer Interaction Interact 03: Ifip Tc13 International Conference on Human-Computer Interaction* (2003), pp. 702–712
15. L. Sweeney, k-anonymity: a model for protecting privacy. Int. J. Uncertain. Fuzziness Knowl. Based Syst. **10**(05), 557–570 (2002). https://doi.org/10.1142/S0218488502001648
16. M. Gruteser, D. Grunwald, Anonymous usage of location-based services through spatial and temporal cloaking, in *Proceedings of the 1st International Conference on Mobile Systems, Applications and Services* (2003), pp. 31–42
17. H. Kido, Y. Yanagisawa, T. Satoh, Protection of location privacy using dummies for location-based services, in *21st International Conference on Data Engineering Workshops (ICDEW'05)* (2005), pp. 1248–1248. https://doi.org/10.1109/ICDE.2005.269
18. L. P. Cox, A. Dalton, V. Marupadi, Smokescreen: flexible privacy controls for presence-sharing, in: *Proceedings of the 5th International Conference on Mobile Systems, Applications and Services* (2007), pp. 233–245. https://doi.org/10.1145/1247660.1247688

19. M. Gruteser, X. Liu, Protecting privacy, in continuous location-tracking applications. IEEE Secur. Priv. **2**(2), 28–34 (2004). https://doi.org/10.1109/MSECP.2004.1281242
20. J. Meyerowitz, R. Roy Choudhury, Hiding stars with fireworks: location privacy through camouflage, in *Proceedings of the 15th Annual International Conference on Mobile Computing and Networking* (2009), pp. 345–356
21. N. Shen, J. Yang, K. Yuan, C. Fu, C. Jia, An efficient and privacy-preserving location sharing mechanism. Comput. Stand. Interfaces **44**, 102–109 (2016). https://doi.org/10.1016/j.csi.2015.06.001
22. J. Chen, S. Su, X. Wang, Towards privacy-preserving location sharing over mobile online social networks. IEICE Trans. Inf. Syst. **102**(1), 133–146 (2019). https://doi.org/10.1587/transinf.2018EDP7187
23. W. Wei, F. Xu, Q. Li, Mobishare: flexible privacy-preserving location sharing in mobile online social networks, in *2012 Proceedings IEEE INFOCOM* (2012), pp. 2616–2620. https://doi.org/10.1109/INFCOM.2012.6195664
24. J. Li, J. Li, X. Chen, Z. Liu, C. Jia, Mobishare+: security improved system for location sharing in mobile online social networks. J. Internet Serv. Inf. Secur. **4**(1), 25–36 (2014)
25. Z. Liu, J. Li, X. Chen, J. Li, C. Jia, New privacy-preserving location sharing system for mobile online social networks, in *2013 Eighth International Conference on P2P, Parallel, Grid, Cloud and Internet Computing* (2013), pp. 214–218
26. Z. Liu, D. Luo, J. Li, X. Chen, C. Jia, N-mobishare: new privacy-preserving location-sharing system for mobile online social networks. Int. J. Comput. Math. **93**(2), 384–400 (2016). https://doi.org/10.1080/00207160.2014.917179
27. G. Sun, Y. Xie, D. Liao, H. Yu, V. Chang, User-defined privacy location-sharing system in mobile online social networks. J. Netw. Comput. Appl. **86**, 34–45 (2017). https://doi.org/10.1016/j.jnca.2016.11.024
28. J. Li, H. Yan, Z. Liu, X. Chen, X. Huang, D.S. Wong, Location-sharing systems with enhanced privacy in mobile online social networks. IEEE Syst. J. **11**(2), 439–448 (2015). https://doi.org/10.1016/j.ins.2020.06.0265
29. C. Xu, X. Xie, L. Zhu, K. Sharif, C. Zhang, X. Du, M. Guizani, PPLS: a privacy-preserving location-sharing scheme in mobile online social networks. Sci. China Inf. Sci. **63**(3), 1–11 (2020). https://doi.org/10.1007/s11432-019-1508-6
30. X. Xiao, C. Chen, A.K. Sangaiah, G. Hu, R. Ye, Y. Jiang, Cenlocshare: a centralized privacy-preserving location-sharing system for mobile online social networks. Future Gener. Comput. Syst. **86**, 863–872 (2018). https://doi.org/10.1016/j.future.2017.01.035

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.