Efficient and Privacy-Preserving Proximity Detection Schemes for Social Applications

Hui Zhu, *Member, IEEE*, Fengwei Wang, Rongxing Lu, *Senior Member, IEEE*, Fen Liu, Gang Fu, and Hui Li, *Member, IEEE*

***Abstract*—With the pervasiveness of location-aware mobile terminals and the popularity of social applications, location-based social networking service (LBSNS) has brought great convenience to people's life. Meanwhile, proximity detection, which makes LBSNS more flexible, has aroused widespread concern. However, the prosperity of LBSNS still faces many severe challenges on account of users' location privacy and data security. In this paper, we propose two efficient and privacy-preserving proximity detection schemes, named arbitrary geometric range query for polygons (AGRQ-P) and arbitrary geometric range query for circles (AGRQ-C), for location-based social applications. With proposed schemes, a user can choose any area on the map, and query whether her/his friends are within the region without divulging the query information to both social application servers and other users, meanwhile, the accurate locations of her/his friends are also confidential for the servers and the query user. Specifically, with algorithms based on ciphertext of geometric range query, users' query and location information is blurred into ciphertext in client, thus no one but the user knows her/his own sensitive information. Detailed security analysis shows that various security threats can be defended. In addition, the proposed schemes are implemented in an IM APP with a real LBS dataset, and extensive simulation results over smart phones further demonstrate that AGRQ-P and AGRQ-C are highly efficient and can be implemented effectively.**

***Index Terms*—Geometric range query, location-based social networking service (LBSNS), privacy-preserving, proximity detection.**

Manuscript received April 21, 2017; revised August 29, 2017; accepted October 17, 2017. Date of publication October 26, 2017; date of current version August 9, 2018. The work of H. Zhu was supported in part by the National Natural Science Foundation of China under Grant 61672411 and Grant U1401251, in part by the National Key Research and Development Program of China under Grant 2017YFB0802201, in part by the Natural Science Basic Research Plan in Shaanxi Province of China under Grant 2016JM6007, and in part by the China 111 Project under Grant B16037. The work of R. Lu was supported in part by the Natural Sciences and Engineering Research Discovery under Grant Rgpin 04009, in part by the NBIF Start-Up under Grant Nbif Rif 2017-915012, in part by the URF under Grant Urf Nf-2017-05, and in part by the HMF under Grant Hmf 2017 Ys-4. An earlier version of this paper was presented at the 13th EAI International Conference on Security and Privacy in Communication Networks, 2017 [1]. *(Corresponding author: Hui Zhu.)*

H. Zhu, F. Wang, F. Liu, and H. Li are with the State Key Laboratory of Integrated Services Networks, Xidian University, Xi'an 710071, China (e-mail: zhuhui@xidian.edu.cn; xdwangfengwei@gmail.com; liufenxd@163.com; lihui@mail.xidian.edu.cn).

R. Lu is with the Faculty of Computer Science, University of New Brunswick, Fredericton, NB E3B 5A3, Canada (e-mail: rlu1@unb.ca).

G. Fu is with Beijing Lanxum Technology Corporation, Beijing 100192, China (e-mail: fugang@lanxum.com).

The implementation of the proposed two schemes and relevant information can be downloaded at http://xdzhuhui.com/demo/AGRQ.

Digital Object Identifier 10.1109/JIOT.2017.2766701

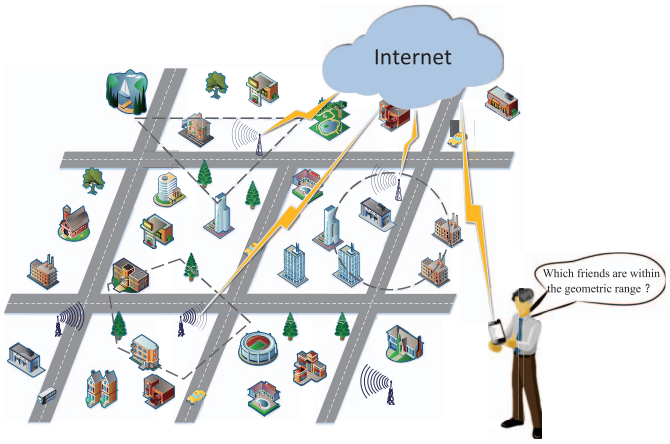


Fig. 1. Conceptual architecture of proximity detection.

I. Introduction

In RECENT years, location-based service (LBS), a general service for mobile devices [1]–[4], has been applied to many areas such as social applications, financial services, transportation, tourism, healthcare, automation, and so on [5]. With the development of social applications, location-based social networking service (LBSNS) has attracted considerable interest. Meanwhile, as a high level location-based function, proximity detection allows users to choose specified geometric range (such as *triangles, circles, and rectangles*) on the map and query which friends of her/his are in the region, as shown in Fig. 1. Proximity detection with geometric range query has been one of the most popular features of LBSNS [6]–[8].

However, the flourish of the LBSNS system still faces severe challenges due to the sensitivity of users' location information [9]–[15]. Once users' sensitive information is compromised, it may lead to computer-assisted crime (harassment, car theft, kidnapping, etc.). Therefore, when users use social applications (such as Wechat, Facebook, Twitter, and so on) for location query, they cannot obtain other users' accurate location information, and their sensitive query information cannot be leaked either. Nevertheless, most LBSNSs rely on the fact that users provide accurate location for service providers, and then service providers provide LBSNS for them. Thus, how to provide accurate LBSNS query results without divulging users' sensitive information to both social application servers (SSs) and other users has become a hot spot of LBS research.

In order to protect the sensitive information of users and solve problems mentioned above, many security techniques have been proposed, such as *k-anonymity* model [16], [17], spatial cloaking techniques [18]–[20], and traditional homomorphic encryption techniques [21]–[24]. Specifically, *k-anonymity* model requires that the anonymous region where the user resides should contain at least other $k-1$ users. The locations of k users are indistinguishable, so that attackers cannot determine the accurate locations of k users. This model can ensure that the probability of obtaining a user's true identity is not greater than $1/k$. But there is a fatal weakness of *k-anonymity*: if k users are in the same location or a sensitive area, such as a hospital, their location information may also be leaked. And in general, *k-anonymity* needs a trusted anonymity server to cloak the location information. Spatial cloaking technique is generally used in privacy protection. The main idea of spatial cloaking technique is that a user's exact location will be masked into a cloaked area which meets the privacy requirements of the user. With spatial cloaking techniques, users' privacy can be well protected, but it brings great communication overhead. Homomorphic encryption is a widely used privacy-preserving technique in proximity detection. In general, it requires complicated arithmetical operations. Since the computation complexity of homomorphic encryption technique is heavy, mobile terminals may not have enough resources to do these operations. These above-mentioned privacy-preserving techniques can protect users' privacy in some degree, but they are not very suitable for mobile terminals.

In this paper, aiming at these above challenges, we propose two efficient privacy-preserving proximity detection schemes for social applications, named arbitrary geometric range query for polygons (AGRQ-P) and arbitrary geometric range query for circles (AGRQ-C), for polygon range query and circle range query, respectively. Specifically, main contributions of this paper are as follows.

1) The proposed schemes can provide privacy-preserving proximity detection for mobile users. With AGRQ-P and AGRQ-C, users' query and location privacy can be well protected. Before being sent to a server, a user's query information and accurate location information are transformed into ciphertext, thus the server of a social application and other users cannot obtain any sensitive information of the user. Apart from this, based on social applications (such as Wechat, Facebook, Twitter, and so on), only registered and authenticated users are allowed to login, which prevents an attacker from disguising a legitimate user to do a query.
2) The proposed schemes can provide accurate query services for users. Based on improving an efficient and privacy-preserving cosine similarity computing protocol [25], we propose two geometric range query algorithms for proximity detection, named GRQ-P and GRQ-C. The proposed algorithms can provide high-precision spatial query while protecting users' privacy.
3) The proposed schemes are efficient in terms of computation complexity and communication overhead. In order to evaluate the effectiveness of our schemes, we develop a demo application, and test through smart phones and workstation with a real dataset. Extensive results show that AGRQ-P and AGRQ-C are effective in the real environment.

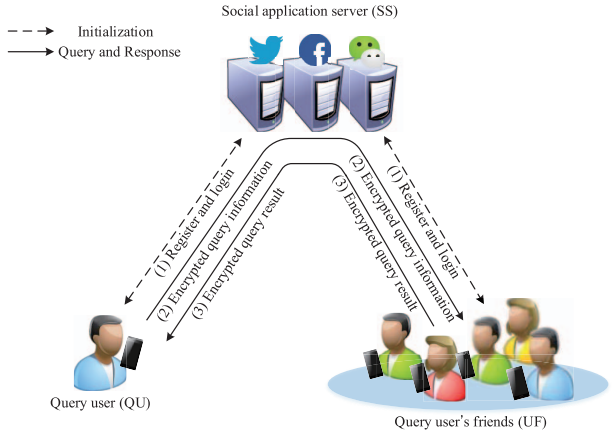


Fig. 2. System model under considered.

The remainder of this paper is organized as follows. In Section II, we formalize the system model, security requirements, and identify our design goal. In Section III, we review the efficient and privacy-preserving cosine similarity computing protocol and the strategy of convex point in polygon as the preliminaries. Then, we propose our arbitrary privacy-preserving geometric range query schemes for proximity detection in Section IV, followed by the security analysis and performance evaluation in Sections V and VI, respectively. We also review some related works in Section VII. Finally, we draw the conclusion in Section VIII.

II. Models and Design Goal

In this section, we formalize the system model and security requirements, and identify our design goal.

A. System Model

The key point of our system design is that a user's sensitive information (such as query range and accurate location information) cannot be obtained by both SS and other users. Specifically, our system consists of three parts: 1) SS; 2) query user (QU); and 3) query user's friends (UFs), as shown in Fig. 2.

1) SS is the server of a social application, which provides users with various of services including LBSNS. After registered in SS, users are allowed to query approximate locations of their friends with LBSNS. In our system, SS is responsible for forwarding data among users and protecting the integrity of data.
2) QU is a user who has already registered in SS. Based on social applications, QU can generate her/his friend list. Then she/he can choose any geometric range on the map, and query which friends of her/his are within the selected region.
3) UF are online friends of QU. In the process of geometric range query, UF receive blurred query from QU, then,

each UF does a hybrid calculation with the blurred query data and her/his own position coordinate to obtain query results, which can only be analyzed by QU with further calculating. Since most calculations are done in client, the computational efficiency of our privacy-preserving schemes should be guaranteed.

B. Security Requirement

Ensuring the privacy of QU's query information and UF's accurate location is crucial for the success of secure proximity detection. In our security model, we consider that SS is credible-but-greedy and QU and UF are honest-but-curious. Specifically, SS will not be fraudulent, but want to get the sensitive information of users from query requests and result responses. QU and UF will not send false information; however, both of them want to obtain each other's sensitive information through the blurred data. Meanwhile, attackers may tamper and modify the data, or impersonate a legitimate user for querying. Considering above security issues, the following security requirements should be satisfied.

1) *Privacy:* Protecting user's query and location information privacy from SS and other users. Specifically, during the query process, QU's geometric query range cannot be obtained by SS and UF, and UF's accurate location information cannot be leaked to QU and SS. In addition, the privacy requirements also include query results, i.e., only the legal QU can decrypt them.
2) *Authentication:* Authenticating that an encrypted query is really sent by a legal QU and not modified during the transmission, i.e., if an illegal user forges a query, this malicious operation should be detected. Moreover, only correct queries can be received by UF. Meanwhile, responses from UF should also be authenticated, so that QU can receive authentic and reliable query results.

C. Design Goal

Under the aforementioned system model and security requirements, our design goal is to develop efficient and privacy-preserving proximity detection schemes with accurate results for social applications. Specifically, the following three objectives should be achieved.

1) *Security and Privacy-Preserving Should Be Guaranteed:* Protecting security and privacy of users' data is the primary goal of the system design. If the proposed schemes do not consider the security, users' query and location information would be divulged. Then, the LBSNS application cannot step into its flourish. Thus, AGRQ-P and AGRQ-C should achieve the confidentiality and authentication simultaneously.
2) *Accuracy of Geometric Query Results Should Be Guaranteed:* Users' experience is an crucial aspect of the proposed schemes, and it is important that the precision of the geometric range query cannot be lowered while protecting users' privacy. Therefore, the proposed schemes should also guarantee high precision.
3) *Low Computation Complexity and Communication Overhead Should Be Achieved:* Although the performance of smart phones is continuously improved today, their batteries are still very limited. In the proposed two schemes, the improvement in computational efficiency can reduce the energy consumption. As a result, AGRQ-P and AGRQ-C should consider the effectiveness in terms of computation and communication to reduce the power consumption of smart phones.

III. Preliminaries

In this section, we review the efficient and privacy-preserving cosine similarity computing protocol [25] and cross products (point in convex polygon strategies) [26]. These will serve as the basis of our schemes.

A. Efficient and Privacy-Preserving Cosine Similarity Computing Protocol

Given a vector of P_A, $\vec{a} = (a_1, a_2, \ldots, a_n) \in F_q^n$ and a vector of P_B, $\vec{b} = (b_1, b_2, \ldots, b_n) \in F_q^n$, we can directly calculate the cosine similarity $\cos(\vec{a}, \vec{b})$ in an efficient and privacy-preserving way. The main calculation process is as follows.

Step 1: (Performed by P_A) Given security parameters k_1, k_2, k_3, k_4, choose two large primes α, p such that $|p| = k_1$, $|\alpha| = k_2$, set $a_{n+1} = a_{n+2} = 0$. Choose a large random $s \in Z_p$ and $n+2$ random numbers $|c_i| = k_3$, $i = 1, 2, \ldots, n+2$. Then P_A calculates

$$C_i = \begin{cases} s(a_i \cdot \alpha + c_i) \bmod p, & a_i \neq 0 \\ s \cdot c_i \bmod p, & a_i = 0 \end{cases}$$

and $A = \sum_{i=1}^{n} a_i^2$. What is more, P_A should keep $s^{-1} \bmod p$ secret. After these operations, $<\alpha, p, C_1, \ldots C_{n+2}>$ will be sent to P_B.

Step 2: (Performed by P_B) Set $b_{n+1} = b_{n+2} = 0$, random numbers $|r_i| = k_4$, then calculate

$$D_i = \begin{cases} b_i \cdot \alpha \cdot C_i \bmod p, & b_i \neq 0 \\ r_i \cdot C_i \bmod p, & b_i = 0 \end{cases}$$

$B = \sum_{i=1}^{n} b_i^2$ and $D = \sum_{i=1}^{n+2} D_i \bmod p$. Then send $<B, D>$ back to P_A.

Step 3: (Performed by P_A) Compute $E = s^{-1} \cdot D \bmod p$, $\vec{a} \cdot \vec{b} = \sum_{i=1}^{n} a_i \cdot b_i = ([E - (E \bmod \alpha^2)]/\alpha^2)$ and $\cos(\vec{a}, \vec{b}) = (\vec{a} \cdot \vec{b}/\sqrt{A} \cdot \sqrt{B})$.

During the above calculation, it can be figured that the vectors of P_A and P_B are confidential to each other.

B. Cross Products—Point in Convex Polygon Strategies

Given a convex polygon P with n edges and a point p, the vertices $P_1P_2 \ldots P_n$ are named in anticlockwise direction. Assume that the coordinates of the vertices and the point are defined as $<(x_1, y_1), (x_2, y_2), \ldots, (x_i, y_i), (x_{i+1}, y_{i+1}), \ldots, (x_n, y_n)>$ and (x_s, y_s), respectively. The point in convex polygon strategy is the protocol to determine whether the point p is within the convex polygon P. We can solve this problem by calculating points orientation [26]. As shown in Fig. 3, the triple points $<P_{i+1}, p, P_i>$ consist of two vertices

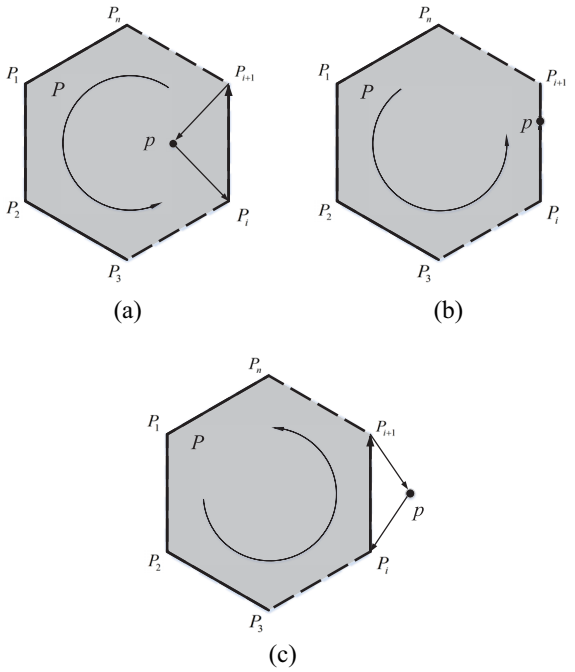


Fig. 3. Orientation of point p and polygon vertexes. (a) Positive orientation. (b) Zero orientation. (c) Negative orientation.

of the polygon and a point p, we defined their orientations as follows.

1) *Positive Orientation:* $<P_{i+1}, p, P_i>$ is a counterclockwise turn.
2) *Negative Orientation:* $<P_{i+1}, p, P_i>$ is a clockwise turn.
3) *Zero Orientation:* $<P_{i+1}, p, P_i>$ is collinear.

The orientation of the $<P_{i+1}, p, P_i>$ can be computed as follows:

$$\begin{aligned} S_i &= \begin{vmatrix} x_{i+1} & y_{i+1} & 1 \\ x_s & y_s & 1 \\ x_i & y_i & 1 \end{vmatrix} \\ &= (x_s \cdot y_i + y_s \cdot x_{i+1} + x_i \cdot y_{i+1}) \\ &\quad - (x_s \cdot y_{i+1} + y_s \cdot x_i + x_{i+1} \cdot y_i). \end{aligned}$$

Next, for the given convex polygon P and point p, whether the point is within the convex polygon can be determined by the following protocol.

1) Let $i \in \{1, 2, \ldots, n\}$, $i' = (i+1) \bmod n$, then compute S_i of the triple points $<P_{i'}, p, P_i>$, in which the vertex P_i is visited in an anticlockwise order.
2) If all $S_i >= 0$, the point p is within the convex polygon P; else, point p is outside the convex polygon P.

IV. Proposed Privacy-Preserving Schemes

Nowadays, the geometric range query is prevalent in proximity detection, especially polygon and circle range query. In this section, based on the above-mentioned preliminaries, we reconstruct the calculation process of traditional point-in-geometric judgement conditions over ciphertext, and design two efficient and privacy-preserving proximity detection schemes, named AGRQ-P and AGRQ-C, for polygon and circle range query, respectively. Both of them mainly consist of two parts: 1) *system initialization* and 2) *privacy-preserving arbitrary geometric range query*. Correspondingly, we propose two efficient and privacy-preserving proximity detection algorithms for them, named GRQ-P (Algorithm 1) and GRQ-C (Algorithm 2), which can be applied to mobile terminals commendably. In addition, Table I is given to show the definition of notations will be used in AGRQ-P and AGRQ-C.

TABLE I
Definition of Notations in AGRQ-P and AGRQ-C

Notation	Definition
k_1, k_2, k_3, k_4	security parameters of privacy-preserving protocol.
α, p	two large primes set by QU.
s, c_{in}, c_i	random numbers used in blurring polygon information.
r_i	random numbers used in hybrid computation.
(x_{qi}, y_{qi})	the *i-th* vertex of QU's query polygon.
(x_q, y_q)	the center of QU's query circle
C	the blurred query data compute by QU.
(x_j, y_j)	the location coordinate of *j-th* UF.
D	the encrypted query response from blurred query data and the coordinate of *j-th* UF.
$E()$	the secure symmetric encryption algorithm.
$E'()$	the secure asymmetric encryption algorithm.
$H()$	the secure cryptographic hash function.

A. System Initialization

SS first chooses an symmetric encryption $E()$, asymmetric encryption algorithm $E'()$, hash function $H()$, and chooses system security parameters k_1, k_2, k_3, and k_4. Then these parameters are applied in the social application.

QU generates its own public and private keys after registered in SS, which are used in bidirectional authentication. Each time QU logs in, a temporary session key k_{QU} is generated through the key negotiation between QU and SS. After this, QU chooses two large primes such that $|p| = k_1$, $|\alpha| = k_2$ and random numbers $s \in Z_p$. Furthermore, QU needs to choose random numbers $|c_k| = k_3$. ($k = in$, i is the number of polygon edges, $n = 1, 2, \ldots, 6$ in AGRQ-P; $k = i = 1, 2, \ldots, 4$ in AGRQ-C).

UF represent online friends of QU. For the sake of simplicity, we first consider that only one friend of QU is online, which is represented by UF_j. During the initialization process, UF_j generates a session key k_{UF_j} with SS, and chooses random numbers $|r_i| = k_4$. (i is the number of polygon edges in AGRQ-P; $i = 3, 4, 5$ in AGRQ-C).

B. Privacy-Preserving Arbitrary Geometric Range Query for Polygons

1) Query Data Creation: Based on social applications, QU chooses vertexes of an convex polygon on the map in anticlockwise order. We assume that the polygon has m edges, and the coordinates of its vertices are as follows:

$$\left\langle (x_{q1}, y_{q1}), (x_{q2}, y_{q2}), \ldots, (x_{qm}, y_{qm}) \right\rangle$$

where the values of coordinates are latitude and longitude with accuracy of two decimal places.

QU chooses two vertices (x_{qi}, y_{qi}) and $(x_{qi'}, y_{qi'})$ to present one edge of the polygon, and executes the following calculation:

$$\begin{aligned} C_{i1} &= s\big(x_{qi} \cdot \alpha + c_{i1}\big) \bmod p \\ C_{i2} &= s\big(y_{qi} \cdot \alpha + c_{i2}\big) \bmod p \\ C_{i3} &= s\big(x_{qi'} \cdot \alpha + c_{i3}\big) \bmod p \\ C_{i4} &= s\big(y_{qi'} \cdot \alpha + c_{i4}\big) \bmod p \\ C_{i5} &= s\big(x_{qi} \cdot y_{qi'} \cdot \alpha + c_{i5}\big) \bmod p \\ C_{i6} &= s\big(x_{qi'} \cdot y_{qi} \cdot \alpha + c_{i6}\big) \bmod p \end{aligned}$$

where $i = 1, 2, \ldots, m$, $i' = (i+1) \bmod m$.

Then QU computes all $C_i = C_{i1} \parallel C_{i2} \parallel C_{i3} \parallel C_{i4} \parallel C_{i5} \parallel C_{i6}$ to obtain $C = C_1 \parallel C_2 \parallel \cdots \parallel C_m$, and creates the message authentication code $\mathrm{MAC}_{\mathrm{QU}} = E_{k_{\mathrm{QU}}}(H(\alpha \parallel p \parallel C \parallel \mathrm{QU} \parallel \mathrm{TS}))$, where TS is current time to resist the potential replay attack.

Finally, QU keeps $s^{-1} \bmod p$ secret, and sends $<\alpha \parallel p \parallel C \parallel \mathrm{QU} \parallel \mathrm{TS} \parallel \mathrm{MAC}_{\mathrm{QU}}>$ to SS.

2) Query Data Transmission: After receiving $<\alpha \parallel p \parallel C \parallel \mathrm{QU} \parallel \mathrm{TS} \parallel \mathrm{MAC}_{\mathrm{QU}}>$, SS first checks TS and $\mathrm{MAC}_{\mathrm{QU}}$ to verify its validity, i.e., verify whether $E_{k_{\mathrm{QU}}}(H(\alpha \parallel p \parallel C \parallel \mathrm{QU} \parallel \mathrm{TS})) = \mathrm{MAC}_{\mathrm{QU}}$. If it does hold, the packet is valid. Then SS computes $\mathrm{MAC}_{SS_q} = E_{k_{\mathrm{UF}_j}}(H(\alpha \parallel p \parallel C \parallel \mathrm{SS} \parallel \mathrm{TS}))$ and sends $<\alpha \parallel p \parallel C \parallel \mathrm{SS} \parallel \mathrm{TS} \parallel \mathrm{MAC}_{SS_q}>$ to UF_j.

3) Response Data Creation: After receiving $<\alpha \parallel p \parallel C \parallel \mathrm{SS} \parallel \mathrm{TS} \parallel \mathrm{MAC}_{SS_q}>$, UF_j checks the time stamp TS and the message authentication code MAC_{SS_q} to verify its validity. Then UF_j computes

$$\begin{aligned} D_{i1} &= r_i \cdot \alpha(x_j \cdot C_{i4} + y_j \cdot C_{i1} + C_{i6}) \bmod p \\ D_{i2} &= r_i \cdot \alpha(x_j \cdot C_{i2} + y_j \cdot C_{i3} + C_{i5}) \bmod p \end{aligned}$$

where $<x_j, y_j>$ is UF_j's location, $i = 1, 2, \ldots, m$.

After that, UF_j computes all $D_i = D_{i1} \parallel D_{i2}$, and makes the order of i chaotic to obtain $D = D_1 \parallel D_2 \parallel \cdots \parallel D_m$. Finally, UF_j creates the message authentication code $\mathrm{MAC}_{UF_j} = E_{k_{UF_j}}(H(D \parallel UF_j \parallel \mathrm{TS}))$, and sends $<D \parallel UF_j \parallel \mathrm{TS} \parallel \mathrm{MAC}_{UF_j}>$ to SS.

4) Response Data Transmission: After receiving $<D \parallel UF_j \parallel \mathrm{TS} \parallel \mathrm{MAC}_{UF_j}>$, SS first checks TS and MAC_{UF_j} to verify its validity. Then SS computes $\mathrm{MAC}_{\mathrm{SS}_a} = E_{k_{\mathrm{QU}}}(H(D \parallel \mathrm{SS} \parallel \mathrm{TS}))$ and returns the query result $<D \parallel \mathrm{SS} \parallel \mathrm{TS} \parallel \mathrm{MAC}_{\mathrm{SS}_a}>$ to QU.

5) Query Results Reading: After receiving $<D \parallel \mathrm{SS} \parallel \mathrm{TS} \parallel \mathrm{MAC}_{\mathrm{SS}_a}>$, QU first checks its validity. Then, QU determines whether UF_j is within the polygon by the following calculations:

$$\begin{aligned} E_{i1} &= s^{-1} \cdot D_{i1} \bmod p \\ &= s^{-1} \cdot r_i \cdot \alpha\big(x_j \cdot C_{i4} + y_j \cdot C_{i1} + C_{i6}\big) \bmod p \\ &= s^{-1} \cdot r_i \cdot s\Big[\alpha^2\big(x_j \cdot y_{qi'} + y_j \cdot x_{qi} + x_{qi'} \cdot y_{qi}\big) \\ &\qquad + \alpha\big(x_j \cdot c_{i4} + y_j \cdot c_{i1} + c_{i6}\big)\Big] \bmod p \\ {E_{i1}}' &= \frac{E_{i1} - E_{i1} \bmod \alpha^2}{\alpha^2} \\ &= r_i\big(x_j \cdot y_{qi'} + y_j \cdot x_{qi} + x_{qi'} \cdot y_{qi}\big) \end{aligned}$$

Algorithm 1 GRQ-P

```
procedure JUDGE(UF_j)        ▷ Whether UF_j is within the
  for i = 1 to i = m do        polygon
    QU computes C_i;
    UF_j computes D_i;
    QU computes E_i;
    if E_i<0 then
      return false;          ▷ UF_j is outside the polygon
    end if
  end for
  return true;               ▷ UF_j is within the polygon
end procedure
```

$$\begin{aligned} E_{i2} &= s^{-1} \cdot D_{i2} \bmod p \\ &= s^{-1} \cdot r_i \cdot \alpha\big(x_j \cdot C_{i2} + y_j \cdot C_{i3} + C_{i5}\big) \bmod p \\ &= s^{-1} \cdot r_i \cdot s\Big[\alpha^2\big(x_j \cdot y_{qi} + y_j \cdot x_{qi'} + x_{qi} \cdot y_{qi'}\big) \\ &\qquad + \alpha\big(x_j \cdot c_{i2} + y_j \cdot c_{i3} + c_{i5}\big)\Big] \bmod p \\ {E_{i2}}' &= \frac{E_{i2} - E_{i2} \bmod \alpha^2}{\alpha^2} \\ &= r_i\big(x_j \cdot y_{qi} + y_j \cdot x_{qi'} + x_{qi} \cdot y_{qi'}\big) \\ E_i &= {E_{i2}}' - {E_{i1}}' \\ &= r_i\big[\big(x_j \cdot y_{qi} + y_j \cdot x_{qi'} + x_{qi} \cdot y_{qi'}\big) \\ &\qquad - \big(x_j \cdot y_{qi'} + y_j \cdot x_{qi} + x_{qi'} \cdot y_{qi}\big)\big]. \end{aligned}$$

Finally, QU computes E_i, $i = 1, 2, \ldots, m$. If all of the $E_i >= 0$, QU can determine that UF_j is within the polygon. Otherwise, UF_j is outside the polygon.

6) Correctness of the GRQ-P: As the calculation presented above, GRQ-P should meet constraints $r_i[\alpha^2(x_j \cdot y_{qi'} + y_j \cdot x_{qi} + x_{qi'} \cdot y_{qi}) + \alpha(x_j \cdot c_{i4} + y_j \cdot c_{i1} + c_{i6})]$, $r_i[\alpha^2(x_j \cdot y_{qi} + y_j \cdot x_{qi'} + x_{qi} \cdot y_{qi'}) + \alpha(x_j \cdot c_{i2} + y_j \cdot c_{i3} + c_{i5})] < p$ and $r_i \cdot \alpha(x_j \cdot c_{i4} + y_j \cdot c_{i1} + c_{i6})$, $r_i \cdot \alpha(x_j \cdot c_{i2} + y_j \cdot c_{i3} + c_{i5}) < \alpha^2$. Since the values of coordinates are not very big, we can choose applicable security parameters easily (such as $k_1 = 512$, $k_2 = 160$, $k_3 = 75$, and $k_4 = 75$). Note that the expression $E_i = r_i[(x_j \cdot y_{qi} + y_j \cdot x_{qi'} + x_{qi} \cdot y_{qi'}) - (x_j \cdot y_{qi'} + y_j \cdot x_{qi} + x_{qi'} \cdot y_{qi})]$, which is formed by two divisors, one is random r_i, and the other one is the cross product of $<P_{i'}, p, P_i>$. Since r_i is a positive number, the sign of the cross product is clear. Then we can find out whether the point is within the polygon through orientations of $<P_{i'}, p, P_i>$, where $i = 1, 2, \ldots, m$.

C. Privacy-Preserving Arbitrary Geometric Range Query for Circles

1) Query Data Creation: QU chooses a center and radius of a circle on the map, which are presented by $<x_q, y_q>$ and r, respectively. $<x_q, y_q>$ is with accuracy of two decimal places, and the minimum value of r is 1 km. Then QU executes the following operations:

$$\begin{aligned} C_1 &= s(x_q \cdot \alpha + c_1) \bmod p \\ C_2 &= s(y_q \cdot \alpha + c_2) \bmod p \\ C_3 &= s \cdot c_3 \bmod p \\ C_4 &= s \cdot c_4 \bmod p. \end{aligned}$$

QU keeps $s^{-1} \bmod p$ secret, and computes $C = C_1 \parallel C_2 \parallel C_3 \parallel C_4$, $A = x_q^2 + y_q^2 - r^2$. Then, QU creates the message authentication code $\mathrm{MAC_{QU}} = E_{k_{\mathrm{QU}}}(H(\alpha \parallel p \parallel A \parallel C \parallel \mathrm{QU} \parallel \mathrm{TS}))$, where TS is current time. After this, QU sends $<\alpha \parallel p \parallel A \parallel C \parallel \mathrm{QU} \parallel \mathrm{TS} \parallel \mathrm{MAC_{QU}}>$ to SS.

2) Query Data Transmission: After receiving $<\alpha \parallel p \parallel A \parallel C \parallel \mathrm{QU} \parallel \mathrm{TS} \parallel \mathrm{MAC_{QU}}>$, SS first checks TS and $\mathrm{MAC_{QU}}$ to verify its validity, i.e., verify whether $E_{k_{\mathrm{QU}}}(H(\alpha \parallel p \parallel A \parallel C \parallel \mathrm{QU} \parallel \mathrm{TS})) = \mathrm{MAC_{QU}}$. If it does hold, the packet is valid. Then, SS computes $\mathrm{MAC}_{SS_q} = E_{k_{\mathrm{UF}_j}}(H(\alpha \parallel p \parallel A \parallel C \parallel \mathrm{SS} \parallel \mathrm{TS}))$, and sends $<\alpha \parallel p \parallel A \parallel C \parallel \mathrm{SS} \parallel \mathrm{TS} \parallel \mathrm{MAC}_{SS_q}>$ to UF_j.

3) Response Data Creation: After receiving $<\alpha \parallel p \parallel A \parallel C \parallel \mathrm{SS} \parallel \mathrm{TS} \parallel \mathrm{MAC}_{SS_q}>$, UF_j first checks its validity. Then UF_j executes the following operations:

$$D_1 = x_j \cdot \alpha \cdot C_1 \bmod p$$
$$D_2 = y_j \cdot \alpha \cdot C_2 \bmod p$$
$$D_3 = r_3 \cdot C_3 \bmod p$$
$$D_4 = r_4 \cdot C_4 \bmod p$$

where $<x_j, y_j>$ is UF_j's location.

After that, UF_j computes $D = r_5 \cdot \sum_{i=1}^{i=4} D_i$ and $B = r_5(x_j^2 + y_j^2 + A)$. Then, UF_j creates the message authentication code $\mathrm{MAC}_{UF_j} = E_{k_{UF_j}}(H(B \parallel D \parallel UF_j \parallel \mathrm{TS}))$ and sends $<B \parallel D \parallel UF_i \parallel \mathrm{TS} \parallel \mathrm{MAC}_{UF_j}>$ to SS.

4) Response Date Transmission: After receiving $<B \parallel D \parallel UF_i \parallel \mathrm{TS} \parallel \mathrm{MAC}_{UF_j}>$, SS first checks its validity. Then, SS computes $\mathrm{MAC}_{\mathrm{SS}_a} = E_{k_{\mathrm{QU}}}(H(B \parallel D \parallel \mathrm{SS} \parallel \mathrm{TS}))$, and sends $<B \parallel D \parallel \mathrm{SS} \parallel \mathrm{TS} \parallel \mathrm{MAC}_{SS_a}>$ to QU.

5) Query Results Reading: After receiving $<B \parallel D \parallel \mathrm{SS} \parallel \mathrm{TS} \parallel \mathrm{MAC}_{SS_a}>$, QU first checks its validity. Then, QU executes the following operations:

$$\begin{aligned} E &= s^{-1} \cdot D \bmod p \\ &= s^{-1} \cdot s \cdot r_5\Big[\alpha^2\big(x_q \cdot x_j + y_q \cdot y_j\big) + \alpha\big(x_j \cdot c_1 + y_j \cdot c_2\big) \\ &\qquad + \ r_3 \cdot c_3 + r_4 \cdot c_4\Big] \bmod p \end{aligned}$$

$$E' = \frac{E - (E \bmod \alpha^2)}{\alpha^2} = r_5(x_q \cdot x_j + y_q \cdot y_j)$$

$$\begin{aligned} R &= B - 2E' \\ &= r_5\Big[x_q^2 + y_q^2 + x_j^2 + y_j^2 - 2(x_q \cdot x_j + y_q \cdot y_j) - r^2\Big] \\ &= r_5\Big[(x_j - x_q)^2 + (y_j - y_q)^2 - r^2\Big]. \end{aligned}$$

Obviously, when $R <= 0$, QU can determine that UF_j is within the polygon. Otherwise, UF_j is outside the polygon.

6) Correctness of the GRQ-C Algorithm: As the calculation presented above, GRQ-C should meet constraints $r_5[\alpha^2(x_q \cdot x_j + y_q \cdot y_j) + \alpha(x_j \cdot c_1 + y_j \cdot c_2) + (r_3 \cdot c_3 + r_4 \cdot c_4)] < p$ and $r_5[\alpha(x_j \cdot c_1 + y_j \cdot c_2) + (r_3 \cdot c_3 + r_4 \cdot c_4)] < \alpha^2$. Since the values of coordinates are not very big, we can choose applicable security parameters easily (such as $k_1 = 512$, $k_2 = 160$, $k_3 = 75$, and $k_4 = 50$). Note that the expression $R = r_5[(x_j - x_q)^2 + (y_j - y_q)^2 - r^2]$, where $(x_j - x_q)^2 + (y_j - y_q)^2$ presents the distance between UF_j and the center of the circle chosen by QU, r_5 is a positive random number. Therefore, whether UF_j is within the circle can be determined by the symbol of R.

Algorithm 2 GRQ-C

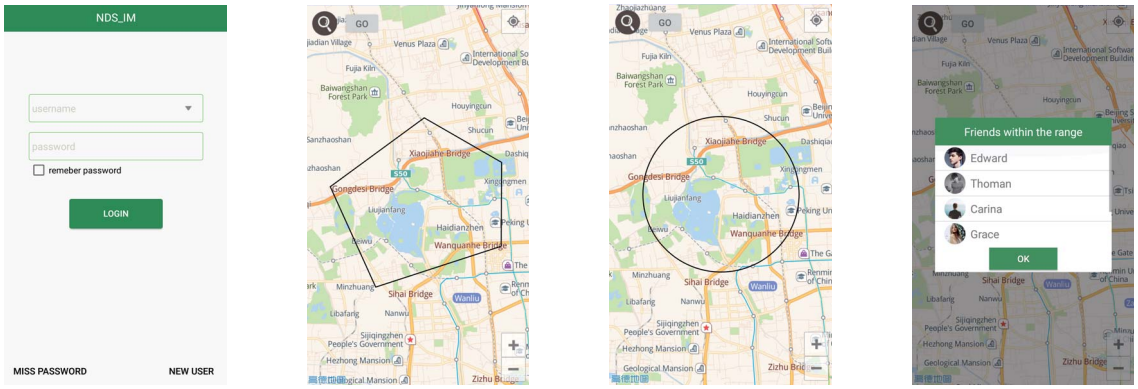
```
procedure JUDGE(UF_j)            ▷ Whether UF_j is within the circle
    QU computes C_i;
    UF_j computes D;
    QU computes R;
    if R_j>0 then
        return false;            ▷ UF_j is outside the circle
    end if
    return true;                 ▷ UF_j is within the circle
end procedure
```

V. Security Analysis

In this section, we analyze the security of the proposed AGRQ-P and AGRQ-C. Specifically, following the security requirements discussed earlier, our analysis will focus on how to preserve the privacy of users, and the authentication during the query process.

A. User's Sensitive Information Is Privacy-Preserving in the Proposed Schemes

1) In AGRQ-P, user's sensitive information consists of two parts: a) the query polygon vertex coordinates of QU and b) location coordinates of UF. With random numbers s and c_{in}, QU transfers the vertexes of the polygon $<(x_{q1}, y_{q1}), (x_{q2}, y_{q2}), \ldots, (x_{qm}, y_{qm})>$ to ciphertext: $C_1 \parallel C_2 \parallel \cdots \parallel C_i \parallel \cdots \parallel C_m$, where $C_i = C_{i1} \parallel C_{i2} \parallel C_{i3} \parallel C_{i4} \parallel C_{i5} \parallel C_{i6}$, and $C_{i1} = s(x_{qi} \cdot \alpha + c_{i1}) \bmod p$, $C_{i2} = s(y_{qi} \cdot \alpha + c_{i2}) \bmod p, \ldots, C_{i6} = s(x_{qi'} \cdot y_{qi} \cdot \alpha + c_{i6}) \bmod p$. Thereby, even if SS and other users are curious about the query information, without knowing the random numbers s and c_{in}, it is impossible to obtain the accurate query information. Moreover, the existence of random numbers c_{in} enhances the space of query information, which can resist the exhaustive attack. Analogously, UF_j computes $D_1 \parallel D_2 \parallel \cdots \parallel D_m$, where $D_i = D_{i1} \parallel D_{i2}$, $D_{i1} = r_i \cdot \alpha(x_j \cdot C_{i4} + y_j \cdot C_{i1} + C_{i6}) \bmod p$ and $D_{i2} = r_i \cdot \alpha(x_j \cdot C_{i2} + y_j \cdot C_{i3} + C_{i5}) \bmod p$. Since UF_j keeps random numbers r_i secret, her/his accurate location coordinate $<x_j, y_j>$ cannot be obtained by SS and QU. And UF_j makes the order of i chaotic, in this way, QU cannot infer the location relationship between UF_j and any edge of the polygon she/he chose on the map. Furthermore, the values of polygon vertex coordinates are with accuracy of two decimal places, which guarantee that the distance between two polygon vertexes is at least 1 km, thus QU cannot infer the accurate locations of UF by choosing multiple overlapping polygons or small range polygons. In addition, even if attackers can capture users' data, they still cannot achieve available information.



(a) (b) (c) (d)

Fig. 4. Implementation of AGRQ-P and AGRQ-C. (a) Register and login. Set (b) polygon with coordinates and (c) circle with center and radius. (d) Query results.

2) In AGRQ-C, user's sensitive information is constituted by the query circle's center coordinate of QU, and location coordinates of UF. With random numbers s and c_i, QU transfers center of the circle $<x_q, y_q>$ to the form: $C_1 \parallel C_2 \parallel C_3 \parallel C_4$, where $C_1 = s(x_q \cdot \alpha + c_1) \bmod p$, $C_2 = s(y_q \cdot \alpha + c_2) \bmod p$, $C_3 = s \cdot c_3 \bmod p$, and $C_4 = s \cdot c_4 \bmod p$. Thereby, without knowing the random numbers s and c_i, it is impossible for SS and other users to obtain the accurate query information. UF_j computes $D = r_5(D_1 + D_2 + \cdots + D_4)$ and $B = r_5(x_j^2 + y_j^2 + A)$, where $D_1 = x_j \cdot \alpha \cdot C_1 \bmod p$, $D_2 = y_j \cdot \alpha \cdot C_2 \bmod p$, $D_3 = r_3 \cdot C_3 \bmod p$, and $D_4 = r_4 \cdot C_4 \bmod p$. Since random numbers r_i are only known by UF_j, her/his accurate location coordinate $< x_j, y_j >$ cannot be obtained by SS and QU. Moreover, D_3 and D_4 ensure that at least two random numbers are included in D, which can prevent QU from guessing UF_j's accurate location. Furthermore, the value of center is with accuracy of two decimal places, and radius r is 1 km, minimum, which guarantee that the distance between two centers is at least 1 km. Thus, QU cannot infer accurate locations of UF by choosing multiple overlapping circles or small range circles. Meanwhile, attackers cannot achieve useful information even if they can capture users' data.

From the above analysis, we can conclude that the user's query information and accurate location can be well protected in AGRQ-P and AGRQ-C.

B. Authentication Is Achieved in the Proposed Schemes

In the proposed two schemes, each registered QU generates her/his own public and private keys. When QU logs in, bidirectional authentication and key negotiation will be performed between QU and SS. Therefore, it is impossible for an attacker to disguise a legitimate QU to forge a geometric range query request. In addition, with proposed schemes, users' encrypted data are verified with message authentication code in each communication between users and SS. In conclusion, if any attacker modifies the data, the action should be detected and resisted.

VI. Performance Evaluation

In this section, we first evaluate the performance of the proposed AGRQ-P and AGRQ-C in terms of computation complexity of QU and UF. Then we implement the proposed two schemes and deploy them in the real environment to evaluate their integrated performance.

A. Evaluation Environment

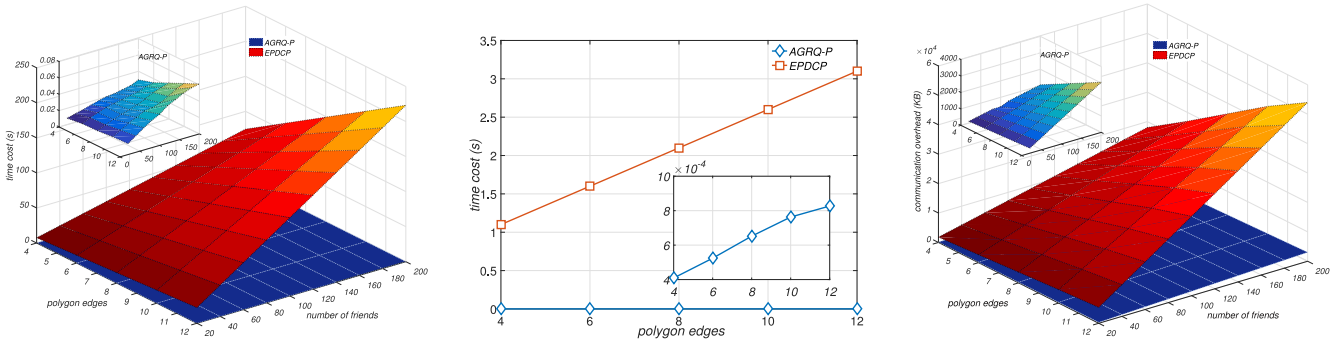
In order to measure the comprehensive performance in the real environment, we implement the proposed schemes in smart phones and workstation. Specifically, smart phones with 2.2-GHz eight-core processor, 3-GB RAM, Android 6.0 and a workstation with 2.0-GHz six-core processor, 64-GB RAM, Ubuntu are chosen to evaluate QU, UF, and SS, respectively, which are connected through 802.11g WLAN. Based on proposed schemes, we construct a social application and install it on smart phones to evaluate QU and UF, then, we build SS on the workstation. As shown in Fig. 4, QU can register in SS, query her/his friends, and display result in the smartphone. in order to evaluate AGRQ-P and AGRQ-C in the real environment, the street map in Beijing is adopted in our application. Furthermore, for the comparison with our schemes, we select two other proximity detection frameworks [enhanced proximity detection for convex polygons (EPDCP) [22] and CRQP] and implement them with the same evaluation environment.

B. Performance Evaluation of AGRQ-P

1) Computation Complexity: The proposed AGRQ-P scheme can offer efficient proximity detection with polygon range query for LBSNS users, we evaluate AGRQ-P in the computation complexity of QU and UF. Specifically, we assume that the number of query polygon vertexes is N, and QU has M online friends. When masking the polygon vertexes information, QU requires $14N$ multiplication operations. After receiving the query from QU, each UF requires $8N$ multiplication operations in hybrid computation. And it costs $4MN$ multiplication operations for QU to read query results. Denote that the multiplication operation is C_m. Therefore, the total computation complexity of QU and UF are $(14N+4MN)*C_m$ and $8N * C_m$, respectively.

TABLE II
COMPUTATION COMPLEXITY OF AGRQ-P, EPDCP, AGRQ-C, AND CRQP

	AGRQ-P	EPDCP	AGRQ-C	CRQP
QU	$(14N+4MN)*C_m$	$(3N+2M+3MN+4l*MN)*C_e+(8N+4M+6MN+l*MN)*C_m$	$(8+5M)*C_m$	$(8+M)*C_e+(8+4M)*C_m$
UF	$8N*C_m$	$(12N+4l*N+l^2*N+9)*C_m+(4N+4l*N+9)*C_e$	$8*C_m$	$12*C_m+4*C_e$



(a) (b) (c)

Fig. 5. Performance evaluation of AGRQ-P versus EPDCP. Average running time in (a) QU versus EPDCP and (b) UF versus EPDCP. (c) Communication cost of AGRQ-P versus EPDCP.

Different from other time-consumption homomorphic encryption techniques, the proposed GRQ-P algorithm uses lightweight multiparty random masking and polynomial aggregation techniques, it can provide accurate proximity detection results and largely reduce the encryption times for mobile terminals. In the following, for the comparison with AGRQ-P, we select an EPDCPs [22], which adopts the same point in convex polygon strategies as AGRQ-P. Denote that the domain size is measured by l and exponentiation operation is presented by C_e. Therefore, for EPCDP, the computation complexities of QU and UF are $(3N + 2M + 3MN + 4l * MN) * C_e + (8N + 4M + 6MN + l * MN) * C_m$ and $(12N + 4l * N + l^2 * N + 9) * C_m + (4N + 4l * N + 9) * C_e$, respectively.

Table II presents the comparison of AGRQ-P and EPDCP. We can clearly see that our proposed GRQ-P can achieve privacy-preserving proximity detection with low complexity. In Fig. 5(a), we plot the computation overhead in QU varying with different numbers of query polygon edges and QU's friends. From the figure, it can be obviously realized that with the increase of polygon edges and QU's friends, the computation overhead of EPDCP increases hugely, which is much higher than that of our proposed AGRQ-P. In Fig. 5(b), we further plot the average running time in UF varying with the increasing number of search polygon edges from 4 to 12, from the figure, it can be clearly seen that the computation overhead in UF of EPDCP is much higher than that of our proposed AGRQ-P, and increases extremely, which verify the above analysis of computation complexity. In conclusion, our proposed AGRQ-P can achieve better efficiency in terms of computation overhead in QU and UF.

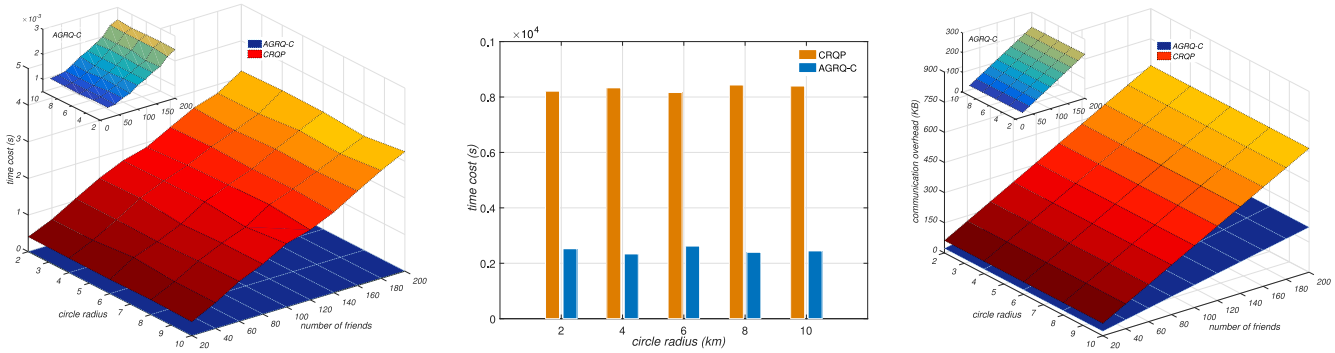
2) Communication Overhead: In AGRQ-P, the query packet is $<\alpha \parallel p \parallel A \parallel \text{QU} \parallel \text{TS} \parallel H_{\text{QU}}>$, and the response packet is $<D_j \parallel UF_j \parallel \text{TS} \parallel H_{SS_A}>$. In the real environment, we record the size of the packets, and compare with EPDCP in one round. As shown in Fig. 5(c), with the increase of the polygon edges and number of QU's friends, the communication overhead of EPDCP significantly increases and it is much higher than that of our proposed AGRQ-P scheme. Although the communication overhead of our proposed AGRQ-P scheme also increases when the numbers of polygon edges and SUs friends are large, it is still much lower than that of EPDCP. In addition, QU needs to interact with UF twice in AGRQ-P, and nine times in EPDCP. In conclusion, our proposed AGRQ-P framework can accomplish better efficiency in terms of communication overhead.

C. Performance Evaluation of AGRQ-C

1) Computation Complexity: The proposed AGRQ-C scheme can offer efficient proximity detection with circle range query for LBSNS users, we evaluate AGRQ-C in the computation complexity of QU and UF. Specifically, we assume that the number of QU's online friends is M. When masking the query circle information, QU requires six multiplication operations. After receiving the query from QU, each UF requires eight multiplication operations in hybrid computation. And it costs $2 + 5M$ multiplication operations for QU to read query results. Therefore, the total computation complexity of QU and UF are $(8 + 5M) * C_m$ and $8 * C_m$, respectively.

In the following, we compare AGRQ-C with CRQP, which is a proximity detection scheme with circle range query based on Paillier's [27] encryption. Based on the above representation, for CRQP, the computation complexities of QU and UF can be obtained, which are $(8 + M) * C_e + (8 + 4M) * C_m$ and $12 * C_m + 4 * C_e$, respectively.

Table II presents the comparison of AGRQ-C and CRQP. It can be clearly seen that our proposed GRQ-C can achieve privacy-preserving proximity detection with low complexity.



(a) (b) (c)

Fig. 6. Performance evaluation of AGRQ-C versus CRQP. (a) Average running time in QU versus CRQP. (b) Average running time in UF versus CRQP. (c) Communication overhead of AGRQ-C versus CRQP.

In Fig. 6(a) and (b), we further plot the computation overhead in QU and UF varying with different number of QU's friends. From the figures, we can obviously find that with the increase of friend numbers, the computation overhead of CRQP increases sharply, which is much higher than that of our proposed AGRQ-C. Although the computation overhead of our proposed AGRQ-C also increases when the number QU's friends is large, it is still much less than that of CRQP. In conclusion, our proposed AGRQ-C can achieve better efficiency in terms of computation overhead in QU and UF.

2) Communication Overhead: In AGRQ-C, the query packet is $<\alpha \parallel p \parallel C \parallel \text{QU} \parallel \text{TS} \parallel H_{\text{QU}}>$, and the response packet is $<B \parallel D \parallel UF_i \parallel \text{TS} \parallel H_{UF_j}>$. In the real environment, we record the size of the packets, and compare with CRQP in one round. Fig. 6(c) shows the communication overhead varying with the radius of query circle and number of QU's friends. From the figure, we can infer that with the increase of QU's friend number, the communication overhead of CRQP significantly increases and it is much higher than that of our proposed AGRQ-C scheme. Although the communication overhead of our proposed AGRQ-C scheme also increases when the number QU's friends is large, it is still much lower than that of CRQP. In conclusion, our proposed AGRQ-C framework can accomplish better efficiency in terms of communication overhead.

VII. Related Work

The study of privacy-preserving spatial query has gained great interest from the research community recently. In this section, we briefly discuss some of them closely related to ours. Many works present privacy-preserving spatial query based on *k-anonymity* [28], [29]. Wang *et al*. [28] presented a new multidimensional *k-anonymity* algorithm based on mapping and divide-and-conquer strategy, whose proposed framework can map the multidimensional to single-dimensional and performs much better than *k-anonymity* in privacy protection. Kalnis *et al.* [29] proposed a framework for preventing location-based identity inference of users who issue spatial queries to location-based services based on *k-anonymity*. The proposed scheme optimized the process of anonymizing the request and processing the transformed spatial queries.

To preserve users' location privacy, spatial cloaking techniques which are based on well-established *k-anonymity* [18]–[20] is frequently used in LBS. Chow *et al.* [18], [19] presented a spatial cloaking algorithm that enables mobile users to obtain location-based services without revealing their exact location information, which is designed for mobile peer-to-peer environment. Wang and Wang [20] proposed an in-device spatial cloaking algorithm which is modified from traditional approaches. Their architecture achieves that spatial cloaking is done on the client side.

Homomorphic encryption techniques are commonly used as methods of blurring privacy information in proximity detection. Yi *et al.* [21] proposed a solution for mobile users to preserve their location and query privacy in approximate *k*-nearest neighbor, the solution is built on the *Paillier* public-key cryptosystem, and can provide both location and query privacy. Mu and Bakiras [22] proposed a novel approach that allows a mobile user to define an arbitrary convex polygon on the map, and test whether her/his friends are within the polygon, which is based on *Paillier* and *ElGamal*. Zhong *et al.* [23] proposed three protocols for location privacy. All of the three protocols are based on *Paillier* cryptosystem, which solve the nearby-friend problem. Thomas [24] proposed a secure point inclusion protocol based on homomorphic encryption, in which the relationship of a point and the polygon is determined by angles. Although homomorphic encryption is widely used, it will bring heavy communication overhead and computation complexity when the number of query points is large.

Nevertheless, in most schemes above, users need to supply their accurate location information for LBSNS providers, which still exists lots of security risks. In order to solve this problem, there are many new location privacy preserving algorithms such as [30] and [31]. Chen and Lin [30] proposed a privacy-preserving point inclusion two-part computation protocol, which based on the relationship of angles formed by vertexes of polygon edges and the point being queried. Similar to our framework, this protocol uses random numbers to blur the location information of users rather than homomorphic encryption. In general, this kind of protocol can reduce the communication overhead and computation complexity. Zhu *et al.* [31] proposed a new secure product protocol,

in which the public-key and third party is not required. This protocol can also be used in lots of privacy-preserving schemes.

Different from the most privacy-preserving schemes which used homomorphic encryption, high time-consuming operations are not required in AGRQ-P and AGRQ-C such as exponentiation operations, pairing operations, and so on. Moreover, spatial cloaking techniques always bring heavy communication overhead, but our two schemes not. In conclusion, the performance of our schemes are better than other similar schemes in the real environment, which has been verified in extensive simulation results.

VIII. Conclusion

In this paper, we have proposed two secure, efficient, and privacy-preserving proximity detection schemes for social applications, called AGRQ-P and AGRQ-C, which proposed new methods for arbitrary geometric range query with improved privacy-preserving cosine similarity computing protocol and point in polygon strategies. The proposed schemes can provide accurate proximity detection results without divulging a user's query and accurate location information to both SSs and other users. Detailed security analysis shows their security strength and privacy-preserving ability, and extensive experiments are conducted to demonstrate their efficiencies.

References

[1] F. Wang *et al.*, "Achieve efficient and privacy-preserving proximity detection scheme for social applications," in *Proc. 13th EAI Int. Conf. Security Privacy Commun. Netw.*, 2017, pp. 1–17.

[2] T. W. Valente, "Network interventions," *Science*, vol. 337, no. 6090, pp. 49–53, 2012.

[3] B. Wang, M. Li, and H. Wang, "Geometric range search on encrypted spatial data," *IEEE Trans. Inf. Forensics Security*, vol. 11, no. 4, pp. 704–719, Apr. 2016.

[4] H. Zhu, F. Liu, and H. Li, "Efficient and privacy-preserving polygons spatial query framework for location-based services," *IEEE Internet Things J.*, vol. 4, no. 2, pp. 536–545, Apr. 2017.

[5] L. Li, R. Lu, and C. Huang, "EPLQ: Efficient privacy-preserving location-based query over outsourced encrypted data," *IEEE Internet Things J.*, vol. 3, no. 2, pp. 206–218, Apr. 2016.

[6] A. Narayanan, N. Thiagarajan, M. Lakhani, M. Hamburg, and D. Boneh, "Location privacy via private proximity testing," in *Proc. NDSS*, 2011, pp. 1–16.

[7] L. Šikšnys, J. R. Thomsen, S. Šaltenis, and M. L. Yiu, "Private and flexible proximity detection in mobile social networks," in *Proc. 11th Int. Conf. Mobile Data Manag.*, Kansas City, MO, USA, 2010, pp. 75–84.

[8] M. Bolic, M. Rostamian, and P. M. Djuric, "Proximity detection with RFID: A step toward the Internet of Things," *IEEE Pervasive Comput.*, vol. 14, no. 2, pp. 70–76, Apr./Jun. 2015.

[9] N. Fei, Y. Zhuang, J. Gu, J. Cao, and L. Yang, "Privacy-preserving relative location based services for mobile users," *China Commun.*, vol. 12, no. 5, pp. 152–161, May 2015.

[10] C. Huang, Z. Yan, N. Li, and M. Wang, "Secure pervasive social communications based on trust in a distributed way," *IEEE Access*, vol. 4, pp. 9225–9238, 2016.

[11] H. Zhu, X. Liu, R. Lu, and H. Li, "Efficient and privacy-preserving online medical prediagnosis framework using nonlinear SVM," *IEEE J. Biomed. Health Inform.*, vol. 21, no. 3, pp. 838–850, May 2017.

[12] B. Wang, M. Li, H. Wang, and H. Li, "Circular range search on encrypted spatial data," in *Proc. IEEE Conf. Commun. Netw. Security (CNS)*, Florence, Italy, 2015, pp. 182–190.

[13] Z. Yan and M. Wang, "Protect pervasive social networking based on two-dimensional trust levels," *IEEE Syst. J.*, vol. 11, no. 1, pp. 207–218, Mar. 2017.

[14] H. Zhu, R. Lu, C. Huang, L. Chen, and H. Li, "An efficient privacy-preserving location-based services query scheme in outsourced cloud," *IEEE Trans. Veh. Technol.*, vol. 65, no. 9, pp. 7729–7739, Sep. 2016.

[15] C. Huang, R. Lu, and H. Zhu, "Privacy-friendly spatial crowdsourcing in vehicular networks," *J. Commun. Inf. Netw.*, vol. 2, no. 2, pp. 59–74, 2017.

[16] L. Sweeney, "k-anonymity: A model for protecting privacy," *Int. J. Uncertainty Fuzziness Knowl. Based Syst.*, vol. 10, no. 5, pp. 557–570, 2002.

[17] B. Gedik and L. Liu, "Protecting location privacy with personalized k-anonymity: Architecture and algorithms," *IEEE Trans. Mobile Comput.*, vol. 7, no. 1, pp. 1–18, Jan. 2008.

[18] C.-Y. Chow, M. F. Mokbel, and X. Liu, "A peer-to-peer spatial cloaking algorithm for anonymous location-based service," in *Proc. 14th Annu. ACM Int. Symp. Adv. Geograph. Inf. Syst.*, Arlington, VA, USA, 2006, pp. 171–178.

[19] C.-Y. Chow, M. F. Mokbel, and X. Liu, "Spatial cloaking for anonymous location-based services in mobile peer-to-peer environments," *GeoInformatica*, vol. 15, no. 2, pp. 351–380, 2011.

[20] S. Wang and X. S. Wang, "In-device spatial cloaking for mobile user privacy assisted by the cloud," in *Proc. 11th Int. Conf. Mobile Data Manag.*, Kansas City, MO, USA, 2010, pp. 381–386.

[21] X. Yi, R. Paulet, E. Bertino, and V. Varadharajan, "Practical approximate k nearest neighbor queries with location and query privacy," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 6, pp. 1546–1559, Jun. 2016.

[22] B. Mu and S. Bakiras, "Private proximity detection for convex polygons," in *Proc. 12th Int. ACM Workshop Data Eng. Wireless Mobile Access*, New York, NY, USA, 2013, pp. 36–43.

[23] G. Zhong, I. Goldberg, and U. Hengartner, "Louis, lester and pierre: Three protocols for location privacy," in *Proc. Int. Workshop Privacy Enhancing Technol.*, Ottawa, ON, Canada, 2007, pp. 62–76.

[24] T. Thomas, "Secure two-party protocols for point inclusion problem," *Int. J. Netw. Security*, vol. 9, no. 1, pp. 1–7, 2009.

[25] R. Lu, H. Zhu, X. Liu, J. K. Liu, and J. Shao, "Toward efficient and privacy-preserving computing in big data era," *IEEE Netw.*, vol. 28, no. 4, pp. 46–50, Jul./Aug. 2014.

[26] F. Feito, J. C. Torres, and A. Ureña, "Orientation, simplicity, and inclusion test for planar polygons," *Comput. Graph.*, vol. 19, no. 4, pp. 595–600, 1995.

[27] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *Proc. Int. Conf. Theory Appl. Cryptograph. Techn.*, Prague, Czechia, 1999, pp. 223–238.

[28] Q. Wang, C. Xu, and M. Sun, "Multi-dimensional k-anonymity based on mapping for protecting privacy," *J. Softw.*, vol. 6, no. 10, pp. 1937–1944, 2011.

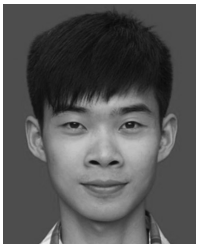
[29] P. Kalnis, G. Ghinita, K. Mouratidis, and D. Papadias, "Preventing location-based identity inference in anonymous spatial queries," *IEEE Trans. Knowl. Data Eng.*, vol. 19, no. 12, pp. 1719–1733, Dec. 2007.

[30] L. Chen and B. Lin, "Privacy-preserving point-inclusion two-party computation protocol," in *Proc. 5th Int. Conf. Comput. Inf. Sci. (ICCIS)*, 2013, pp. 257–260.

[31] Y. Zhu *et al.*, "Fast secure scalar product protocol with (almost) optimal efficiency," in *Proc. Int. Conf. Collaborative Comput. Netw. Appl. Worksharing*, Wuhan, China, 2015, pp. 234–242.

Hui Zhu (M'13) received the B.Sc. degree from Xidian University, Xi'an, China, in 2003, the M.Sc. degree from Wuhan University, Wuhan, China, in 2005, and the Ph.D. degree from Xidian University, in 2009.

In 2013, he was a Research Fellow with the School of Electrical and Electronics Engineering, Nanyang Technological University, Singapore. Since 2016, he has been a Professor with the School of Cyber Engineering, Xidian University. His current research interests include applied cryptography, data security, and privacy.



Fengwei Wang received the B.Sc. degree from Xidian University, Xi'an, China, in 2016, where he is currently pursuing the master's degree at the School of Cyber Engineering.

His current research interests include applied cryptography, cyber security, and privacy.

Rongxing Lu (S'09–M'10–SM'15) received the Ph.D. degree from the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada, in 2012.

He has been an Assistant Professor with the Faculty of Computer Science, University of New Brunswick, Fredericton, NB, Canada, since 2016. He was an Assistant Professor with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore, from 2013 to 2016. He was a Post-Doctoral Fellow with the University of Waterloo, from 2012 to 2013. His current research interests include applied cryptography, privacy enhancing technologies, and Internet of Things–big data security and privacy.

Dr. Lu was a recipient of the Governor General's Gold Medal, and the 8th IEEE Communications Society (ComSoc) Asia–Pacific Outstanding Young Researcher Award in 2013. He currently serves as the Secretary of the IEEE ComSocCIS-TC.

Fen Liu received the B.Sc. and M.Sc. degrees from Xidian University, Xi'an, China, in 2014 and 2017, respectively.

Her current research interests include applied cryptography, cyber security, and privacy.

Gang Fu received the B.Sc. and M.Sc. degrees from Tsinghua University, Beijing, China, in 1992 and 1997, respectively.

Since 2015, he has been the CTO with the Beijing Lanxum Technology Corporation, Beijing, China. His current research interests include computer network, network security, data security, and privacy.

Hui Li (M'10) received the B.Sc. degree from Fudan University, Shanghai, China, in 1990, and the M.Sc. and Ph.D. degrees from Xidian University, Xi'an, China, in 1993 and 1998, respectively.

Since 2005, he has been a Professor with the School of Telecommunication Engineering, Xidian University. His current research interests include cryptography, wireless network security, information theory, and network coding.

Dr. Li served as the TPC Co-Chair of ISPEC 2009 and IAS 2009, the General Co-Chair of E-Forensic 2010, ProvSec 2011, and ISC 2011, and the Honorary Chair of NSS 2014 and ASIACCS 2016.