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 AGRQ-P and 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 are blurred into chipertext 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—Location-based social networking service, proximity detection, privacy-preserving, geometric range query.

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 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, 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 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 AGRQ-P and AGRQ-C, for polygon range query and circle range query respectively. Specifically, main contributions of this paper are as follows.

- **First**, 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.

- **Second**, 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.

- **Third**, 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.

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 section V and section VI, respectively. We also review some related works in section VII. Finally, we draw our conclusions 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 the social application server and other users. Specifically, our system consists of three parts: Social Application Server (SS), Query User (QU) and Query User’s Friends (UF), as shown in Fig.2.

![System model under considered.](image)

- **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.

- **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.

- **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, 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.

- **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.

- **Authentication.** Authenticating that an encrypted query is really sent by a legal QU and not modified during the transmission, i.e., 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.

- **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, AGRO-P and AGRO-C should achieve the confidentiality and authentication simultaneously.

- **Accuracy of geometric query results should be guaranteed.** Users’ experience is a 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.

- **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, AGRO-P and AGRO-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.

Step1: (performed by $P_A$) Given security parameters $k_1$, $k_2$, $k_3$, $k_4$, choose two large primes $\alpha, \rho$ such that $|\rho| = k_1, |\alpha| = k_2$, 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 $C_i = \{s(a_i \cdot c_i + a_{i+1} \cdot c_{i+1}) \mod p, a_{i+1} \neq 0; s \cdot c_i \mod p, a_{i+1} = 0;\}$ and $A = \sum_{i=1}^{n} a_i^2$. What’s more, $P_A$ should keep $s^{-1} \mod p$ secret. After these operations, $< a, p, C_1, \ldots, C_{n+2} >$ will be sent to $P_B$.

Step2: (performed by $P_B$) Set $b_{n+1} = b_{n+2} = 0$, random numbers $|r_i| = k_4$, then calculate $D_i = \{b_i \cdot c_i \mod p, b_i \neq 0; r_i \cdot c_i \mod p, b_i = 0;\}$ $B = \sum_{i=1}^{n} b_i^2$ and $D = \sum_{i=1}^{n+2} D_i \mod p$. Then send $< B, D >$ back to $P_A$.

Step3: (performed by $P_A$) Compute $E = s^{-1} \cdot D \mod p$, $\vec{a} \cdot \vec{b} = \sum_{i=1}^{n} a_i \cdot b_i = \frac{E - (E \mod \alpha)}{\alpha}$ and $\cos(\vec{a}, \vec{b}) = \frac{\vec{a} \cdot \vec{b}}{\sqrt{\sum_{i=1}^{n} a_i^2 \cdot \sum_{i=1}^{n} b_i^2}}$.

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 $(x, y)$, 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 $< P_{i+1}, P_i >$ consist of two vertices of the polygon and a point $p$, we defined their orientations as follows.

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

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

$$S_i = \begin{vmatrix} x_{i+1} & y_{i+1} & 1 \\ x_i & y_i & 1 \\ x & y & 1 \end{vmatrix} = (x \cdot y_i + y \cdot x_{i+1} + x_i \cdot y_{i+1}) - (x \cdot y_{i+1} + y \cdot x_i + x_{i+1} \cdot y_i)$$
A. System Initialization

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

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.

- Let $i \in \{1, 2, \ldots, n\}$, $i' = (i + 1) \mod n$, 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.
- If all $S_i \geq 0$, 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: system initialization and privacy-preserving arbitrary geometric range query. Correspondingly, we propose two efficient and privacy-preserving proximity detection algorithms for them, named GRQ-P and GRQ-C, 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.

B. Privacy-preserving Arbitrary Geometric Range Query for Polygons (AGRQ-P)

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 vertexes are as follows.

$$< (x_{q1}, y_{q1}), (x_{q2}, y_{q2}), \ldots, (x_{qm}, y_{qm}) >$$

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

QU chooses two vertexes $(x_{qi}, y_{qi}), (x_{qj}, y_{qj})$ to present one edge of the polygon, and executes the following calculation.

$$C_{i1} = s(x_{qi} \cdot \alpha + c_{i1}) \mod p$$
$$C_{i2} = s(y_{qi} \cdot \alpha + c_{i2}) \mod p$$
$$C_{i3} = s(x_{qi} \cdot \alpha + c_{i3}) \mod p$$
$$C_{i4} = s(y_{qi} \cdot \alpha + c_{i4}) \mod p$$
$$C_{i5} = s(x_{qi} \cdot y_{qi} \cdot \alpha + c_{i5}) \mod p$$
$$C_{i6} = s(x_{qj} \cdot y_{qj} \cdot \alpha + c_{i6}) \mod p$$

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

TABLE I

<table>
<thead>
<tr>
<th>Notation</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>$k_1, k_2, k_3, k_4$</td>
<td>security parameters of privacy-preserving protocol.</td>
</tr>
<tr>
<td>$\alpha, p$</td>
<td>two large primes set by QU.</td>
</tr>
<tr>
<td>$s_i, c_{i1}, c_{i2}$</td>
<td>random numbers used in blurring polygon information.</td>
</tr>
<tr>
<td>$r_i$</td>
<td>random numbers used in hybrid computation.</td>
</tr>
<tr>
<td>$(x_i, y_i)$</td>
<td>the $i$-th vertex of QU's query polygon.</td>
</tr>
<tr>
<td>$(x, y)$</td>
<td>the center of QU's query circle.</td>
</tr>
<tr>
<td>$C$</td>
<td>the blurred query data compute by QU.</td>
</tr>
<tr>
<td>$(x_i, y_i)$</td>
<td>the location coordinate of $j$-th UF.</td>
</tr>
<tr>
<td>$D$</td>
<td>the encrypted query response from blurred query data and the coordinate of $j$-th UF.</td>
</tr>
<tr>
<td>$E()$</td>
<td>the secure symmetric encryption algorithm.</td>
</tr>
<tr>
<td>$E'(())$</td>
<td>the secure asymmetric encryption algorithm.</td>
</tr>
<tr>
<td>$H()$</td>
<td>the secure cryptographic hash function.</td>
</tr>
</tbody>
</table>
Then QU computes all $C_j = C_{i1} \parallel C_{i2} \parallel C_{i3} \parallel C_{i4} \parallel C_{i5} \parallel C_{i6}$ to obtain $C = C_1 \parallel C_2 \parallel \cdots \parallel C_m$, and creates the message authentication code $MAC_{QU} = E_{k_{qu}}(H(\alpha \parallel p \parallel C \parallel QU \parallel TS))$, where $TS$ is current time to resist the potential replay attack.

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

**Query Data Transmission.** After receiving $< \alpha \parallel p \parallel C \parallel QU \parallel TS \parallel MAC_{QU} >$, SS first checks $TS$ and $MAC_{QU}$ to verify its validity, i.e., verify whether $E_{k_{qu}}(H(\alpha \parallel p \parallel C \parallel QU \parallel TS)) = MAC_{QU}$. If it does hold, the packet is valid. Then SS computes $MAC_{SS_{q}} = E_{k_{sp}}(H(\alpha \parallel p \parallel C \parallel SS \parallel TS))$ and sends $< \alpha \parallel p \parallel C \parallel SS \parallel TS \parallel MAC_{SSq} >$ to $UF_j$.

**Response Data Creation.** After receiving $< \alpha \parallel p \parallel C \parallel SS \parallel TS \parallel MAC_{SSq} >$, $UF_j$ checks the time stamp $TS$ and the message authentication code $MAC_{SSq}$ to verify its validity. Then $UF_j$ computes

$$D_{i1} = r_1 \cdot x(C_4 + y \cdot C_1 + C_6) \mod p$$
$$D_{i2} = r_1 \cdot x(C_4 + y \cdot C_3 + C_5) \mod p$$

, where $<x_i,y_i>$ is $UF_j$’s location, $i = 1, 2, \ldots, m$.

After that, $UF_j$ computes all $D_{i1} \parallel D_{i2} \parallel \cdots \parallel D_m$. Finally, $UF_j$ creates the message authentication code $MAC_{UF_j} = E_{k_{uf}}(H(D \parallel UF_j \parallel TS))$, and sends $< D \parallel UF_j \parallel TS \parallel MAC_{UF_j} >$ to SS.

**Response Data Transmission.** After receiving $< D \parallel UF_j \parallel TS \parallel MAC_{UFj} >$, SS first checks $TS$ and $MAC_{UF_j}$ to verify its validity. Then SS computes $MAC_{SSq} = E_{k_{sp}}(H(D \parallel SS \parallel TS))$ and returns the query result $< D \parallel SS \parallel TS \parallel MAC_{SSq} >$ to QU.

**Query Results Reading.** After receiving $< D \parallel SS \parallel TS \parallel MAC_{SSq} >$, QU first checks its validity. Then, QU determines whether $UF_j$ is within the polygon by following calculations.

$$E_{i1} = s^{-1} \cdot D_{i1} \mod p$$
$$= s^{-1} \cdot r_1 \cdot x(C_4 + y \cdot C_1 + C_6) \mod p$$
$$= s^{-1} \cdot r_1 \cdot x[s(\alpha^2(C_4 + y \cdot C_1 + C_6) + \alpha(C_4 + y \cdot C_3 + C_5) \mod p$$
$$E_{i1}' = \frac{E_{i1} - E_1 \mod p}{\alpha^2}$$
$$= r_1(C_4 + y \cdot C_1)$$

$$E_{i2} = s^{-1} \cdot D_{i2} \mod p$$
$$= s^{-1} \cdot r_1 \cdot x(C_4 + y \cdot C_3 + C_5) \mod p$$
$$= s^{-1} \cdot r_1 \cdot x[s(\alpha^2(C_4 + y \cdot C_3 + C_5) + \alpha(C_4 + y \cdot C_1 + C_6) \mod p$$
$$E_{i2}' = \frac{E_{i2} - E_2 \mod p}{\alpha^2}$$
$$= r_1(C_4 + y \cdot C_1)$$

$$E_i = E_{i2}' - E_{i1}'$$

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.

**Algorithm 1 GRQ-P**

```plaintext```
procedure JUDGE(UFj)  
    for i = 1 to m do  
        QU computes Ci;  
        UFj computes Di;  
        QU computes Ei;  
        if Ei < 0 then  
            return false;  
        end if  
    end for  
    return true;  
end procedure
```

**Correctness of the GRQ-P.** As the calculation presented above, GRQ-P should meet constraints $r_1[\alpha^2(C_4 + y \cdot C_1 + C_6) + \alpha(C_4 + y \cdot C_3 + C_5)] < p$ and $r_1[\alpha^2(C_4 + y \cdot C_1 + C_6) + \alpha(C_4 + y \cdot C_3 + C_5)] > \alpha^2$. 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 $E_i = r_1(C_4 + y \cdot C_1)$ is within the polygon. Otherwise, $< P_r, P, P_1 >$. Since $r_1$ 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 $< P_r, P, P_1 >$, where $i = 1, 2, \ldots, m$.

C. Privacy-preserving Arbitrary Geometric Range Query for Circles (AGRQ-C)

**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 1km. Then QU executes the following operations.

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

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

2327-4662 (c) 2017 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.
Algorithm 2 GRQ-C

```
procedure Judge(UFj)  // Whether UFj is within the circle
  QU computes Cj;
  UFj computes D;
  QU computes R;
  if R > 0 then
    return false;  // UFj is outside the circle
  end if
  return true;  // UFj is within the circle
end procedure
```

Correctness of the GRQ-C algorithm. As the calculation presented above, GRQ-C should meet constraints
where \( x_j \) and \( y_j \) represent the coordinates of the query point \( J \).

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. The user's sensitive information is privacy-preserving in the proposed schemes.

- In AGRQ-P, user’s sensitive information consists of two parts: the query polygon vertex coordinates of QU, and 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: \( C_1 \parallel C_2 \parallel \cdots \parallel C_{m}, \) where \( C_1 = C_{i1} \parallel C_{i2} \parallel C_{i3} \parallel C_{i4} \parallel C_{i5} \parallel C_{i6}, \) and \( C_{i1} = s(x_{q1} \cdot \alpha + c_{i1}) \mod p, C_{i2} = s(y_{q1} \cdot \alpha + c_{i2}) \mod p, \ldots, C_{i6} = s(x_{qm} \cdot y_{qm} \cdot \alpha + c_{i6}) \mod p. \) 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, \( UFj \) computes \( D_1 \parallel D_2 \parallel \cdots \parallel D_{m}, \) where \( D_1 = D_{i1} \parallel D_{i2}, D_{i1} = r_i \cdot (x_{j1} \cdot C_{i4} + y_{j1} \cdot C_{i3} + C_{i6}) \mod p \) and \( D_{i2} = r_i \cdot (x_{j1} \cdot C_{i2} + y_{j1} \cdot C_{i3} + C_{i6}) \mod p. \) Since \( UFj \) keeps random numbers \( r_i \) secret, her/his accurate location coordinate \( < x_j, y_j > \) cannot be obtained by SS and QU. And \( UFj \) makes the order of \( i \) chaotic, in this way, QU cannot infer the location relationship between \( UFj \) 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 \( 1km \), 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.

- 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 \( C_1 = s(x_q \cdot \alpha + c_i) \mod p, C_2 = s(y_q \cdot \alpha + c_2) \mod p, C_3 = s \cdot c_3 \mod p \) and \( C_4 = s \cdot c_4 \mod p. \) Therefore, without knowing the random numbers \( s \) and \( c_{i} \), it is impossible for SS and other users to obtain the accurate query information. \( UFj \) computes \( D = r_5(D_1 + D_2 + \cdots + D_4) \) and \( B = r_5(x_j^2 + y_j^2 + A), \) where \( D_1 = x_j \cdot (x_q \cdot c_1 + y_q \cdot c_2) + (r_3 \cdot c_3 + r_4 \cdot c_4) < \alpha^2. \) 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 \( UFj \) and the center of the circle chose by QU, \( r_5 \) is a positive random number. Therefore, whether \( UFj \) is within the circle can be determined by the symbol of \( R. \)
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, 3GB 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 (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 \((14N + 4MN) \times C_m \) and \( 8N \times C_m \), respectively.

Different from other time-consumption homomorphic encryption techniques, the proposed GRQ-P algorithm uses lightweight multi-party 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 a enhanced proximity detection for convex polygons (EPDCP) [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 EPDCP, the computation complexities of QU and UF are \((3N + 2M + 3MN + 4l \times MN) \times C_e \) + \((8N + 4M + 6MN + l \times MN) \times C_m \) and \((12N + 4l \times N + l^2 \times N + 9) \times C_e + (4N + 4l \times N + 9) \times C_m \), 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 $< \alpha \parallel p \parallel A \parallel QU \parallel TS \parallel H_{QU}>$, and the response packet is $< D_j \parallel UF_j \parallel TS \parallel H_{S_{j}} >$. 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 SU’s 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 6 multiplication operations. After receiving the query from QU, each UF requires 8 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 $(8 + 5M) \times C_m$ and $8 \times C_m$, respectively.

In the following, we compare AGRQ-C with CRQP, which is a proximity detection scheme with circle range query based on Pallier [27] encryption. Based on the above representation, for CRQP, the computation complexities of QU and UF can be obtained, which are $(8 + M) \times C_e + (8 + 4M) \times C_m$ and $12 \times C_m + 4 \times C_e$, 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. 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 $< \alpha \parallel p \parallel C \parallel QU \parallel TS \parallel H_{QU} >$, and the response packet is $< B \parallel D \parallel UF_j \parallel TS \parallel H_{E_{F}} >$. 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 multi-dimensional 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 et al. [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. Xu et al. [21] proposed a solution for mobile users to preserve their location and query privacy in approximate $k$-nearest neighbor (KNN), the solution is built on the Paillier public-key cryptosystem, and can provide both location and query privacy. Mu et al. [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 the 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 et al. [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], [31]. Li et al. [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

---

(a) Average running time in QU vs CRQP.

(b) Average running time in UF vs CRQP.

(c) Communication overhead of AGRQ-C vs CRQP.

Fig. 6. Performance evaluation of AGRQ-C vs CRQP.
point in polygon strategies. The proposed schemes can pro-
provide accurate proximity detection results without divulging a
user’s query and accurate location information to both social
application servers and other users. Detailed security analysis
shows their security strength and privacy-preserving ability,
and extensive experiments are conducted to demonstrate their
efficiencies.

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

References
efficient and privacy-preserving proximity detection scheme for social
applications,” in 13th EAI International Conference on Security and
spatial data,” IEEE Transactions on Information Forensics and Security,
spatial query framework for location-based services,” IEEE Internet of
based query over outsourced encrypted data,” IEEE Internet of Things
“Location privacy via private proximity testing,” in NDSS, 2011.
flexible proximity detection in mobile social networks,” in 2010 Eleventh
International Conference on Mobile Data Management. IEEE, 2010,
pp. 75–84.
[8] M. Bolic, M. Rostamian, and P. M. Djuric, “Proximity detection with
rdf: a step toward the internet of things,” IEEE Pervasive Computing,
vol. 14, no. 2, pp. 70–76, 2015.
[9] N. Fei, Y. Zhuang, J. Gu, J. Cao, and L. Yang, “Privacy-preserving rel-
ative location based services for mobile users,” China Communications,
vol. 12, no. 5, pp. 152–161, 2015.
communications based on trust in a distributed way,” IEEE Access,
online medical prediagnosis framework using nonlinear svm,” IEEE
journal of biomedical and health informatics, vol. 21, no. 3, pp. 838–
850, 2017.
encrypted spatial data,” in Communications and Network Security (CNS),
preserving location based services query scheme in outsourced cloud,”
IEEE Transactions on Vehicular Technology, vol. 65, no. 9, pp. 7729–
7739, Sept 2016.
in vehicular networks,” Journal of Communications and Information
[16] L. Sweeney, “k-anonymity: A model for protecting privacy,” Inter-
national Journal of Uncertainty, Fuzziness and Knowledge-Based Systems,
vol. 10, no. 05, pp. 557–570, 2002.
[17] B. Gedik and L. Liu, “Protecting location privacy with personalized k-
anonymity: Architecture and algorithms,” IEEE Transactions on Mobile
[18] C.-Y. Chow, M. F. Mokbel, and X. Liu, “Peer-to-peer spatial cloaking
algorithm for anonymous location-based service,” in Proceedings of the
14th annual ACM international symposium on Advances in geographic

Hui Zhu (M13) received his B.Sc. degree from Xi-
dian University in 2003, M.Sc. degree from Wuhan
University in 2005, and Ph.D. degrees from Xidian
University in 2009. In 2013, he was with School of
Electrical and Electronics Engineering, Nanyang
Technological University as a research fellow.
Since 2016, he has been the professor in the
School of Cyber Engineering, Xidian University,
China. His research interests include the areas of
applied cryptography, data security and privacy.

Fengwei Wang received the B.Sc. degree from
Xidian University, Xi’an, China, in 2016. He is
currently working toward the Master’s degree with
the School of Cyber Engineering, Xidian University,
Xi’an, China.
His research interests include the areas of
applied cryptography, cyber security, and privacy.
Rongxing Lu (S09-M10-SM15) has been an assistant professor at the Faculty of Computer Science, University of New Brunswick (UNB), Canada, since August 2016. Before that, he worked as an assistant professor at the School of Electrical and Electronic Engineering, Nanyang Technological University (NTU), Singapore from April 2013 to August 2016. Rongxing Lu worked as a Postdoctoral Fellow at the University of Waterloo from May 2012 to April 2013. He was awarded the most prestigious "Governor Generals Gold Medal", when he received his PhD degree from the Department of Electrical and Computer Engineering, University of Waterloo, Canada, in 2012; and won the 8th IEEE Communications Society (ComSoc) Asia Pacific (AP) Outstanding Young Researcher Award, in 2013. He is presently a senior member of IEEE Communications Society.

His research interests include applied cryptography, privacy enhancing technologies, and IoT-Big Data security and privacy. Dr. Lu currently serves as the Secretary of 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 research interests include the areas of applied cryptography, cyber security, and privacy.

Gang Fu received his B.Sc. degree from Tsinghua University in 1992, M.Sc. degree from Tsinghua University in 1997. Since 2015 he has been the CTO of Lanxum Technology Corporation. His research interests include the areas of computer network, network security, data security and privacy.

Hui Li (M10) received his B.Sc. degree from Fudan University in 1990, M.Sc. and Ph.D. degrees from Xidian University in 1993 and 1998, respectively. Since 2005, he has been the professor in the school of Telecommunication Engineering, Xidian University, China. His research interests are in the areas of cryptography, wireless network security, information theory and network coding.