Achieve Efficient and Privacy-Preserving Proximity Detection Scheme for Social Applications

Fengwei Wang[1,4], Hui Zhu[1](✉), Rongxing Lu[2], Fen Liu[1], Cheng Huang[3], and Hui Li[1]

[1] State Key Laboratory of Integrated Services Networks,
Xidian University, Xi'an, China
zhuhui@xidian.edu.cn
[2] Faculty of Computer Science, University of New Brunswick, Fredericton, Canada
[3] Department of Electrical and Computer Engineering, University of Waterloo,
Waterloo, Canada
[4] Science and Technology on Communication Networks Laboratory,
Shijiazhuang, China

Abstract. This paper proposes an efficient scheme, named CPSS, to perform privacy-preserving proximity detection based on chiphertext of convex polygon spatial search. We consider a scenario where users have to submit their location and search information to the social application server for accessing proximity detection service of location-based social applications (LBSAs). With proximity detection, users can choose any polygon area on the map and search whether their friends are within the select region. Since the location and search information of users are sensitive, submitting these data over plaintext to the social application server raises privacy concerns. Hence, we propose a novel method, with which users can access proximity detection without divulging their search and location information. Specifically, the data of a user is blurred into chipertext in client, thus no one can obtain the sensitive information except the user herself/himself. We prove that the scheme can defend various security threats and validate our scheme using a real LBS dataset. Also, we show that our proposed CPSS is highly efficient in terms of computation complexity and communication overhead.

Keywords: Location-based social application · Proximity detection
Privacy-preserving · Convex polygon spatial search

1 Introduction

Nowadays, with the flourish of the location-based service (LBS) and social networking, location-based social applications (LBSAs) have attracted considerable interest. These applications enormously benefit people in a variety of contexts ranging from their work to personal life. For example, when a individual is traveling in a strange place, LBSAs can help her/him meet with friends in the

X. Lin et al. (Eds.): SecureComm 2017, LNICST 238, pp. 339–355, 2018.
https://doi.org/10.1007/978-3-319-78813-5_17

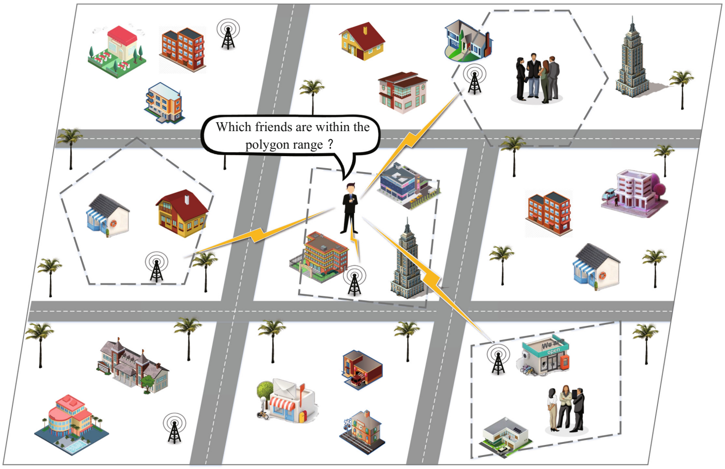


Fig. 1. Conceptual architecture of proximity detection.

surroundings [1–4]. Proximity detection is a high level location based service of LBSAs, which enables a user to choose any range on the map, and search which friends of her/his are within this region, as shown in Fig. 1. Proximity detection with polygon spatial search has been one of the most popular features of LBSAs [5–9].

Although LBSAs benefit people by providing convenient lifestyle, its development still faces severe challenges due to the sensitivity of users' location information [10–16]. For example, users' sensitive information could be analyzed or revealed by LBSAs server easily. Once these sensitive information is obtained by attackers, mobile users may be harmed economically, physically, and legally. Therefore, when users use social applications (such as Wechat, Facebook, Twitter and so on) for location search, their sensitive search and location information cannot be leaked. However, most LBSAs rely on the fact that users submit accurate location over plaintext to the social application server, then the server provides LBS for them. Thus, how to provide accurate LBS search results without divulging users' sensitive information has become a hot spot of LBS research.

Aiming at these above challenges, in this paper, we propose an efficient and privacy-preserving proximity detection scheme for social applications, named CPSS. Specifically, main contributions of this paper are as follows.

- *First*, the proposed CPSS provides a privacy-preserving proximity detection framework for LBSAs. With CPSS, a user can keep her/his search and location information secret from social application servers and other users. Specifically, in our novel CPSS scheme, users' search and their location data are transformed into chipertext with random masking technique in client, thus social application servers cannot obtain any sensitive information of users. Meanwhile, no one but the user knows her/his own sensitive information.

Moreover, based on social applications, users are authenticated when login, therefore, it is impossible for an attacker to disguise a legitimate user to execute a search.

- *Second*, the proposed CPSS provides accurate spatial search service for users. We construct an convex polygon spatial search algorithm based on improving an efficient and privacy-preserving cosine similarity computing protocol [17], named PSS, which can provide high-precision convex polygon spatial search while protecting users' privacy.
- *Third*, CPSS provides proximity detection service in the real environment efficiently. We evaluate the performance of the proposed CPSS in terms of the computation complexity and communication overhead, and deploy CPSS in smart phones and workstation with a real LBS dataset. Extensive experiment results demonstrate that CPSS is highly effective in the real environment.

The rest of this paper is organized as follows: we review the related works in Sect. 2. The efficient and privacy-preserving cosine similarity computing protocol and the strategy of point in convex polygon are reviewed as the preliminaries in Sect. 3. In Sect. 4, we formalize the models, design goal, and propose our privacy-preserving proximity detection scheme with convex polygon spatial search followed the security analysis of the proposed scheme in Sect. 5. Performance evaluation in Sect. 6. Conclusions are discussed in Sect. 7.

2 Related Work

The field of privacy-preserving spatial search has witnessed several different techniques those have been proposed to protect users' privacy ([18–22] and reference therein). In this section, we review some of them resumptively.

K-anonymity [23] is a traditional technique to perform privacy-preserving spatial search. There are few works have been proposed in this direction [18,19, 24]. In 2011, [18] presented a new multidimensional *k-anonymity* algorithm based on mapping and divide-and-conquer strategy. The work in [18] maps the multi-dimensional to single-dimensional and performs much better than *k-anonymity* in privacy protection. In [19], Sharma and Shen et al. utilized the *k-anonmity* mechanism with an entropy factor to check the possible probability of detecting a subscriber in a region by an adversary based on previous traces. In their work, they aimed to maximize the entropy based on a random mobility pattern before generating a new cloaking region. Gedik and Liu et al. [24] proposed a location privacy architecture which use a flexible privacy personalization framework to support location *k-anonymity* for a wide range of mobile clients with context-sensitive privacy requirements. This framework enables each mobile client to specify the minimum level of anonymity. However, *k-anonymity* requires that the anonymous region where the user resides should contain at least other k-1 users, if k users are in the same location, their location information may also be leaked, and it brings heavy communication overhead to users.

Spatial cloaking technique is widely used to ensure users' privacy through masking the user accurate location into a cloaked spatial regions [20,21,25]. The schemes in [21] proposed to enable mobile users to obtain location-based services without revealing their exact location by designing a spatial cloaking algorithm, which is suitable for mobile peer-to-peer environment. In 2015, [20] proposed a new spatial cloaking technique to hide a user's location with a cloaking of the serving based station. Different from the most existing approaches, the work in [20] selects a properly chosen dummy location from real locations of $eNodeBs$ to minimize side information for an adversary. Wang et al. proposed an in-device spatial cloaking algorithm in [25] to achieve processing data in client. The work in [25] is modified from traditional approaches. However, in general, the schemes using spatial cloaking technique return a list of candidate search answers instead of the exact answer, which brings heavy communication overhead to users.

In order to mitigate the heavy communication overhead involved with the *k-anonmity* and spatial cloaking, homomorphic encryption technique is commonly used to achieve privacy-preserving spatial search. In 2016, [22] proposed a solution for mobile users to preserve their location and query privacy in approximate k nearest neighbor (KNN). The work in [22] is built on the *Paillier* public-key cryptosystem, and can provide both location and query privacy security. In [26], Mu and Bakiras proposed a novel privacy-preserving spatial query approach using *Paillier* and *ElGamal*. In their work, a mobile user is allowed to define an arbitrary convex polygon on the map, and test whether her/his friends are within the polygon. The methods in [27] proposed for secure distance computation over encrypted data, in their work, the underlying security is ensured by the homomorphic encryption scheme which support computation on encrypted data. Thomas et al. using homomorphic encryption proposed a secure point inclusion protocol in [28]. They determined the relationship of a point and the polygon by angles. Nevertheless, most homomorphic encryption schemes require massive resource-consuming computation, which brings heavy computation complexity.

These above-mentioned schemes are not very suitable for mobile devices. Hence, in this paper, we use a new but lightweight technique to construct an efficient and privacy-preserving proximity detection scheme with convex polygon spatial search. Our approach is highly efficient in terms of computation complexity and communication overhead. Most importantly, our scheme doesn't reduce the search accuracy due to the privacy-preserving requirements.

3 Preliminaries

Recently Lu et al. [17] proposed an efficient and privacy-preserving cosine similarity computing protocol and in 1995, Feito et al. [29] proposed the cross product (point in convex polygon strategies). In this section, we review theses as the basis of our scheme.

3.1 Efficient and Privacy-Preserving Cosine Similarity Computing Protocol

Given a vector of P_A, $\mathbf{a} = (a_1, a_2, \ldots, a_n) \in F_q^n$ and a vector of P_B, $\mathbf{b} = (b_1, b_2, \ldots, b_n) \in F_q^n$, we can directly calculate the cosine similarity $\cos(\mathbf{a}, \mathbf{b})$

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 α, p such that $|p| = k_1$, $|\alpha| = k_2$, set $a_{n+1} = a_{n+2} = 0$. Choose a large random $s \in \mathbb{Z}_p^*$ 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's 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.

Step2: (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 $D = \sum_{i=1}^{n+2} D_i \bmod p$. After this P_B sends $<B, D>$ back to P_A.

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

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

3.2 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 vertexes 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 cross product is the protocol to determine whether the point p is within the convex polygon P. We can solve this problem by calculating points orientation [29]. As shown in Fig. 2, the triple points $<P_{i+1}, p, P_i>$ consist of two vertices of the convex polygon and a point p, we defined their orientations as follows.

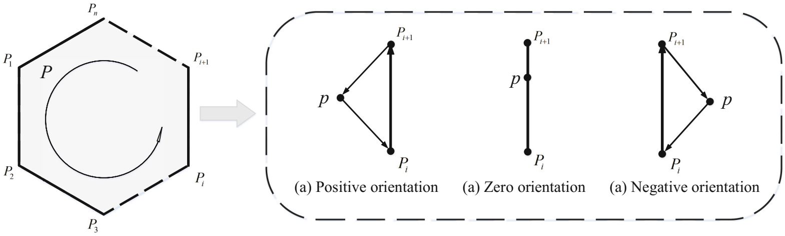


Fig. 2. Orientation of point p and polygon vertex.

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

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

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

4 Models, Design Goal and Proposed CPSS Scheme

4.1 Models and Design Goal

Let us consider that the system model consists of three parts: *Social Application Server* (SS), *Search User* (SU) and *Search User's Friends* (UF), as shown in Fig. 3.

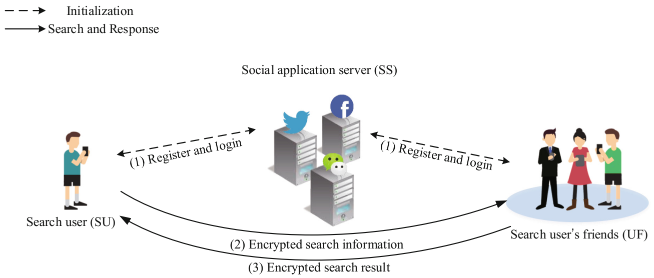

Initialization
Search and Response
Social application server (SS)
(1) Register and login
(1) Register and login
Search user (SU)
Search user's friends (UF)
(2) Encrypted search information
(3) Encrypted search result

Fig. 3. System model under considered.

- We consider a server of a LBSA as SS, which provides users with various of services including proximity detection. Users registered in SS are allowed to search approximate location of their friends with proximity detection. In our system, SS is responsible for forwarding data among users and protecting the integrity of data.
- A user who wants to execute a search and has already registered in SS is represented by SU. Based on social applications, SU can generate her/his friend list. Then she/he can choose any polygon range on the map, and search which friends of her/his are within the selected region.

- UF present online friends of SU. In the process of polygon spatial search, UF receive blurred search information from SU, then each UF does a hybrid calculation with the blurred search data and her/his own position coordinate to obtain search results, which can only be analyzed by SU with further calculating. Since most calculations are done in client, the computational efficiency of our privacy-preserving scheme should be guaranteed.

Within the system model, let us introduce the threat model and define the following security requirements of the proposed work. In the threat model, we consider that SS is credible-but-greedy, SU and UF are honest-but-curious. Specifically, SS will not be fraudulent, but want to get the sensitive information of users from search requests and result responses. SU and UF will not send false information, nevertheless, 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 to execute a search. Considering above security issues, the following security requirements should be satisfied.

- *Privacy.* On one hand, protecting user's search and location information secret from SS and other users, even if SS can obtain all requests and responses from users, it still cannot identity user's search polygon range and location information accurately. On the other hand, the privacy requirements also include search results, i.e., only the legal SU can decrypt them.
- *Authentication.* Authenticating that an encrypted proximity detection search request is really sent by a legal SU and not modified during the transmission, i.e., if an illegal user forges a search, this malicious operation should be detected. That is, only correct search requests can be received by UF, meanwhile, responses from UF should also be authenticated, so that SU can receive the reliable search results.

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

- *Security should be guaranteed.* Once the security of the proposed is not achieved, users' sensitive information (i.e., search and location data) could be disclosed, which may harm users severely. In this way, it is hard for LBSA to step into its flourish. Therefore, achieving the confidentiality and authentication simultaneously is the primary goal of CPSS.
- *Accuracy of polygon search results should be guaranteed.* It is significant that applying the privacy-preserving strategy cannot compromise the accuracy. Therefore, the proposed framework should also provide the same search result as that of the scheme unusing privacy-preserving technique.
- *Low computation complexity and communication overhead should be achieved.* Considering the batteries of mobile devices are very limited today. The proposed scheme should enhance the computational efficiency to reduce the energy consumption in mobile devices. As a result, CPSS should have low overhead in terms of computation and communication.

4.2 Proposed CPSS Scheme

The proposed efficient and privacy-preserving proximity detection scheme mainly consist of two parts: *system initialization* and *privacy-preserving convex polygon spatial search*. Detailed explanation is as follows.

System Initialization. SS first chooses system security parameters p_1, p_2, p_3, p_4, a secure symmetric encryption $E()$, i.e., AES, a secure asymmetric encryption algorithm $E'()$, i.e., ECC and a secure hash function $H()$. Then SS generates its private key sk_{SS} and public key pk_{SS}. SS keeps sk_{SS} secret, and publishes the system parameters $<E(), E'(), H(), p_1, p_2, p_3, p_4, pk_{SS}>$.

When registering in SS, SU sets her/his *password*, and generates private key sk_{SU} and public key pk_{SU}. When logging in, SU is authorized with $<password, SU, pk_{SS}, E'()>$. Meanwhile, a temporary session key k_{SU} is generated through the key negotiation. The authentication scheme of register and login for social applications is sophisticated [30]. After this, SU chooses two large primes such that $|\beta| = p_1$, $|\alpha| = p_2$, a large random number $u \in \mathbb{Z}_p^*$ and random numbers $|v_{in}| = p_3$, where i is the number of polygon edges, $n = 1, 2, \cdots, 6$.

UF represent online friends of SU. For the sake of simplicity, we first consider that only one friend of SU 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 $|w_i| = p_4$, where i is the number of polygon edges.

Privacy-Preserving Convex Polygon Spatial Search. At the beginning, we design the PSS algorithm for the proposed scheme, which mainly consists of three functions: *SearchGeneration*, *ResultGeneration* and *ResultReading*. The description of the functions is as follows.

- $SearchGeneration(\alpha, \beta, u, V, D)$: The function takes as input two big primes α and β, random number u, an array V with elements v_{in} and vertexes of the search polygon D. It outputs the blurred data of the search polygon, which is presented by Q. Assume that the vertexes of the polygon are $<(x_{q1}, y_{q1}), (x_{q2}, y_{q2}), \ldots, (x_{qm}, y_{qm})>$ in anticlockwise order. Detailed calculations of this function are as follows.

$$\begin{aligned}
&Q = Q_1 \parallel Q_2 \parallel \cdots \parallel Q_i \parallel \cdots \parallel Q_m\\
&Q_i = Q_{i1} \parallel Q_{i2} \parallel Q_{i3} \parallel Q_{i4} \parallel Q_{i5} \parallel Q_{i6}\\
&Q_{i1} = u(x_{qi} \cdot \alpha + v_{i1}) \bmod \beta\\
&Q_{i2} = u(y_{qi} \cdot \alpha + v_{i2}) \bmod \beta\\
&Q_{i3} = u(x_{qi'} \cdot \alpha + v_{i3}) \bmod \beta\\
&Q_{i4} = u(y_{qi'} \cdot \alpha + v_{i4}) \bmod \beta\\
&Q_{i5} = u(x_{qi} \cdot y_{qi'} \cdot \alpha + v_{i5}) \bmod \beta\\
&Q_{i6} = u(x_{qi'} \cdot y_{qi} \cdot \alpha + v_{i6}) \bmod \beta,
\end{aligned}$$

where $i = 1, 2, ..., m$, $i' = (i + 1) \bmod m$. The process is conducted by SU, after this, the data of the search polygon are blurred into chipertext Q.

- *ResultGeneration*(α, β, W, Q, C): This function is executed by UF_j. It outputs the search result R with the inputs α, β, W, Q and C, where α and β are two big primes, W is an array with elements w_i, Q is the blurred polygon information and C is the location coordinate of UF_j. Assume that the location coordinate of UF_j is $<x_j, y_j>$. Specific computing process is as follows.

$$\begin{aligned} R &= R_1 \parallel R_2 \parallel \cdots \parallel R_i \parallel \cdots \parallel R_m \\ R_i &= R_{i1} \parallel R_{i2} \\ R_{i1} &= w_i \cdot \alpha(x_j \cdot Q_{i4} + y_j \cdot Q_{i1} + Q_{i6}) \bmod \beta \\ R_{i2} &= w_i \cdot \alpha(x_j \cdot Q_{i2} + y_j \cdot Q_{i3} + Q_{i5}) \bmod \beta, \end{aligned}$$

where $i = 1, 2, ..., m$. Note that in the operation $R = R_1 \parallel R_2 \parallel \cdots \parallel R_i \parallel \cdots \parallel R_m$, the order of i should be rearranged. After this process, the search result R is generated, which can only be decrypted by legitimate SU.

- *ResultReading*(β, u, R): This function takes as input the big prime β, random number u and search result R. It decrypts R and outputs the judgement J, which shows whether UF_j is with the polygon area. Concretely, the operations are as follows.

$$\begin{aligned} J_{i1} &= u^{-1} \cdot R_{i1} \bmod \beta \\ &= u^{-1} \cdot w_i \cdot \alpha(x_j \cdot Q_{i4} + y_j \cdot Q_{i1} + Q_{i6}) \bmod \beta \\ &= u^{-1} \cdot w_i \cdot u[\alpha^2(x_j \cdot y_{qi'} + y_j \cdot x_{qi} + x_{qi'} \cdot y_{qi}) \\ &\quad + \alpha(x_j \cdot v_{i4} + y_j \cdot v_{i1} + v_{i6})] \bmod \beta \\ {J_{i1}}' &= \frac{J_{i1} - (J_{i1} \bmod \alpha^2)}{\alpha^2} \\ &= w_i(x_j \cdot y_{qi'} + y_j \cdot x_{qi} + x_{qi'} \cdot y_{qi}) \end{aligned}$$

$$\begin{aligned} J_{i2} &= u^{-1} \cdot R_{i2} \bmod \beta \\ &= u^{-1} \cdot w_i \cdot \alpha(x_j \cdot Q_{i2} + y_j \cdot Q_{i3} + Q_{i5}) \bmod \beta \\ &= u^{-1} \cdot w_i \cdot u[\alpha^2(x_j \cdot y_{qi} + y_j \cdot x_{qi'} + x_{qi} \cdot y_{qi'}) \\ &\quad + \alpha(x_j \cdot v_{i2} + y_j \cdot v_{i3} + v_{i5})] \bmod \beta \\ {J_{i2}}' &= \frac{J_{i2} - (J_{i2} \bmod \alpha^2)}{\alpha^2} \\ &= w_i(x_j \cdot y_{qi} + y_j \cdot x_{qi'} + x_{qi} \cdot y_{qi'}) \end{aligned}$$

$$\begin{aligned} J_i &= {J_{i2}}' - {J_{i1}}' \\ &= w_i[(x_j \cdot y_{qi} + y_j \cdot x_{qi'} + x_{qi} \cdot y_{qi'}) \\ &\qquad - (x_j \cdot y_{qi'} + y_j \cdot x_{qi} + x_{qi'} \cdot y_{qi})] \end{aligned}$$

For $i = 1, 2, ..., m$, if all of the $J_i > 0$, this function outputs that J is $true$, Otherwise, outputs J is $false$.

Algorithm 1. PSS

```
procedure JUDGE($UF_j$)                    ▷ Whether $UF_j$ is within the polygon
    for $i = 1$ to $i = m$ do
        $SU$ computes $Q$;
        $UF_j$ computes $R$;
        $SU$ computes $J_i$;
        if $J_i <= 0$ then
            return $J$ is $false$;         ▷ $UF_j$ is outside the polygon
        end if
    end for
    return $J$ is $true$;                  ▷ $UF_j$ is within the polygon
end procedure
```

Correctness of the PSS. As the calculation presented above, PSS should meet constraints $w_i \cdot \alpha^2(x_j \cdot y_{qi'} + y_j \cdot x_{qi} + x_{qi'} \cdot y_{qi})$, $w_i \cdot \alpha^2(x_j \cdot y_{qi} + y_j \cdot x_{qi'} + x_{qi} \cdot y_{qi'}) < \beta$ and $\alpha(x_j \cdot v_{i2} + y_j \cdot v_{i3} + v_{i5})$, $\alpha(x_j \cdot v_{i2} + y_j \cdot v_{i3} + v_{i5}) < \alpha^2$. Since the values of coordinates are not very big, we can choose applicable security parameters easily (such as $p_1 = 512$, $p_2 = 160$, $p_3 = 75$ and $p_4 = 75$). Note that the expression $J_i = w_i[(x_j \cdot y_{qi} + y_j \cdot x_{qi'} + x_{qi} \cdot y_{qi'}) - (x_j \cdot y_{qi'} + y_j \cdot x_{qi} + x_{qi'} \cdot y_{qi})]$, which is formed by two divisors, one is random w_i, and the other is the cross product of $<P_{i'}, p, P_i>$. Since w_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 $i = 1, 2, ..., m$.

Next, based on PSS algorithm, we propose the efficient and privacy-preserving proximity detection scheme with convex polygon spatial search, and illustrate it in Fig. 4. The detailed procedure is described as below.

(1) *Generate the search request*: Based on social applications, SU executes the *system initialization* to generate random numbers α, β, u, V, and chooses vertexes of the search polygon D. Then she/he generates the search data Q by calling $SearchGeneration(\alpha, \beta, u, V, D)$, and creates the message authentication code $MAC_{SU} = E_{k_{SU}}(H(\alpha \parallel \beta \parallel Q \parallel SU \parallel TS))$, where TS is current time to resist the potential replay attack. Finally, SU keeps $u^{-1} \bmod \beta$ secret, and sends $<\alpha \parallel \beta \parallel Q \parallel SU \parallel TS \parallel MAC_{SU}>$ to SS.

(2) *Verify the search request and forward*: SS first checks TS and MAC_{SU} to verify the validity of data, i.e., verify whether $E_{k_{SU}}(H(\alpha \parallel \beta \parallel Q \parallel SU \parallel TS)) = MAC_{SU}$. If it does hold, the packet is valid. Then SS computes $MAC_{SS_q} = E_{k_{UF_j}}(H(\alpha \parallel \beta \parallel Q \parallel SS \parallel TS))$, and sends $< \alpha \parallel \beta \parallel Q \parallel SS \parallel TS \parallel MAC_{SS_q}>$ to UF_j.

(3) *Generate the search response*: UF_j checks the time stamp TS and MAC_{SS_q} to verify the validity of data. Then UF_j executes the *system initialization* to generate random numbers W, and generates the search result R by calling $ResultGeneration(\alpha, \beta, W, Q, C)$, where C is the location of UF_j. Finally, UF_j computes $MAC_{UF_j} = E_{k_{UF_j}}(H(R \parallel UF_j \parallel TS))$, and sends $<R \parallel UF_j \parallel TS \parallel MAC_{UF_j}>$ to SS.

(4) *Verify the search response and forward*: SS first checks TS and MAC_{UF_j} to verify the validity of the packet. Then SS computes $MAC_{SS_a} = E_{k_{SU}}(H(R \parallel SS \parallel TS))$, and returns the search result $<R \parallel SS \parallel TS \parallel MAC_{SS_a}>$ to SU.

(5) *Read the search response*: After receiving $<R \parallel SS \parallel TS \parallel MAC_{SS_a}>$, SU first checks its validity, and determines whether UF_j is within the polygon by calling *ResultReading(*β, u, R*)*.

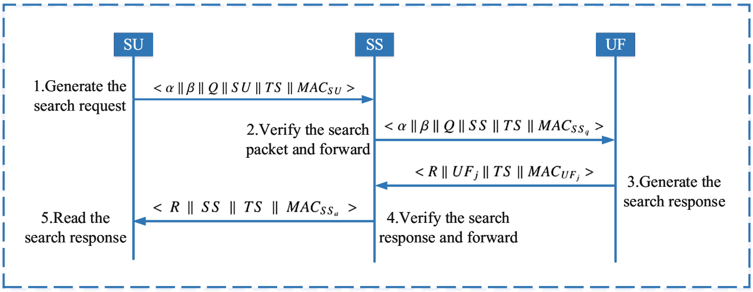

SU
SS
UF
1.Generate the search request
$< \alpha \parallel \beta \parallel Q \parallel SU \parallel TS \parallel MAC_{SU} >$
2.Verify the search packet and forward
$< \alpha \parallel \beta \parallel Q \parallel SS \parallel TS \parallel MAC_{SS_q} >$
$< R \parallel UF_j \parallel TS \parallel MAC_{UF_j} >$
3.Generate the search response
$< R \parallel SS \parallel TS \parallel MAC_{SS_a} >$
4.Verify the search response and forward
5.Read the search response

Fig. 4. Proposed CPSS scheme.

5 Security Analysis

Following the security requirements discussed earlier, in this section, we analysis the security of the proposed CPSS. We will focus on how the proposed CPSS can preserve the privacy of users, and the authentication during the search process.

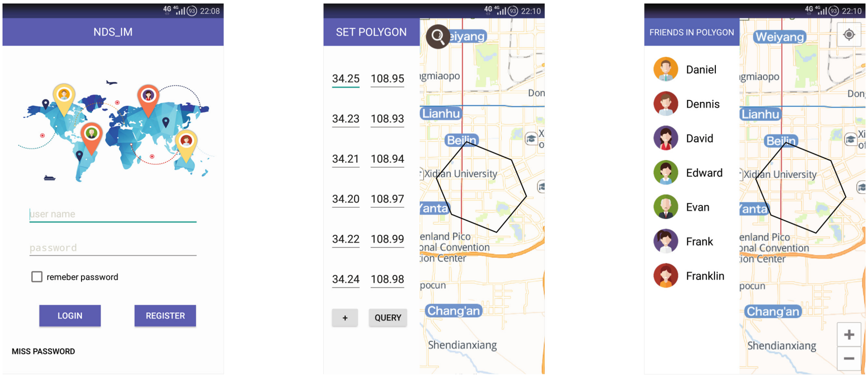
5.1 The User's Sensitive Information is Privacy-Preserving in the Proposed Scheme

In the proposed CPSS, by using random numbers u and v_{in}, the vertexes of the search polygon $<(x_{q1}, y_{q1}), (x_{q2}, y_{q2}), \ldots, (x_{qm}, y_{qm})>$ are encrypted in the form of $Q_1 \parallel Q_2 \parallel \cdots \parallel Q_i \parallel \cdots \parallel Q_m$, where $Q_i = Q_{i1} \parallel Q_{i2} \parallel Q_{i3} \parallel Q_{i4} \parallel Q_{i5} \parallel Q_{i6}$, and $Q_{i1} = u(x_{qi} \cdot \alpha + v_{i1}) \bmod \beta$, $Q_{i2} = u(y_{qi} \cdot \alpha + v_{i2}) \bmod \beta$, $\cdots$, $Q_{i6} = u(x_{qi'} \cdot y_{qi} \cdot \alpha + v_{i6}) \bmod \beta$. Since u and v_{in} are only known by SU, even if SS and other users are curious about the search information, it is impossible for them to obtain the accurate search information. Moreover, the space of search data is increased by random numbers v_{in} to resist the exhaustive attack. Analogously, UF_j computes $R = R_1 \parallel R_2 \parallel \cdots \parallel R_m$ over blurred search data, where $R_i = R_{i1} \parallel R_{i2}$, $R_{i1} = w_i \cdot \alpha(x_j \cdot Q_{i4} + y_j \cdot Q_{i1} + Q_{i6}) \bmod \beta$ and $R_{i2} = w_i \cdot \alpha(x_j \cdot Q_{i2} + y_j \cdot Q_{i3} + Q_{i5}) \bmod \beta$. Since the location coordinate $<x_j, y_j>$ is blurred with random numbers w_i which are only known by UF_j, SS and SU cannot obtain the location coordinate accurately. Moreover, the order of i is rearranged during the operation $R = R_1 \parallel R_2 \parallel \cdots \parallel R_m$, in this way, SU cannot infer the location relationship between UF_j and any edge of the polygon she/he chose on the map. Furthermore, the input values of polygon vertex coordinates are limited with accuracy of two decimal places to guarantee that the distance between two polygon vertexes is at least 1 km, thus SU cannot infer the accurate locations of UF by choosing multiple overlapping polygons or small range polygons on the map. In addition, due to the users' data is all encrypted with random numbers in client, even if attackers can capture users' data, they still cannot achieve available information.

From the above analysis, we can conclude that user's search information and accurate location are secure in the proposed CPSS.

5.2 Authentication is Achieved in the Proposed Scheme

The authentication scheme of register and login for social applications is sophisticated, each registered user generates her/his own private key and its corresponding public key. When the user logs in, mutual authentication and key negotiation will be performed between the user and SS. Therefore, it is impossible for an attacker to disguise a legitimate user to forge a polygon spatial search request. In addition, with the proposed scheme, the message authentication code MAC is computed with the hash function $H()$, and is encrypted with the secure symmetric encryption algorithm $E()$ in each communication between users and SS. Therefore, without knowing the session key k, it is impossible for attackers to modify the data between users and SS. As a result, the search request from the unregistered user and the modified information can be detected in the proposed CPSS.



34.25	108.95
34.23	108.93
34.21	108.94
34.20	108.97
34.22	108.99
34.24	108.98

Daniel
Dennis
David
Edward
Evan
Frank
Franklin

(a) Register and login. (b) Set polygon with coordinates. (c) Show search results.

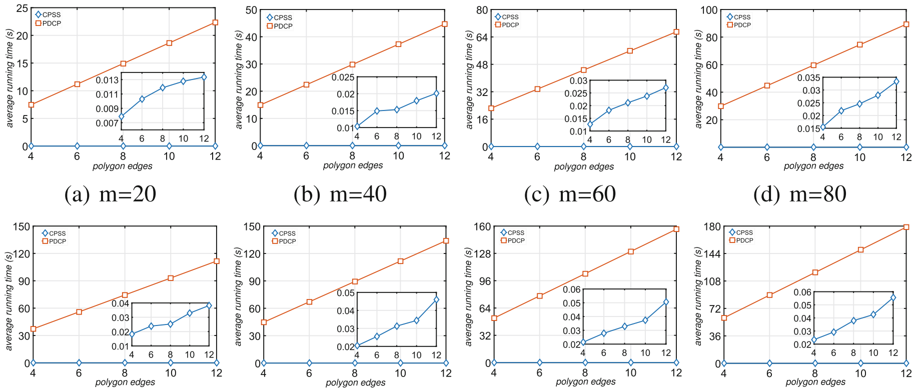
Fig. 5. Implementation of CPSS.

6 Performance Evaluation

In this section, we demonstrate the performance of our scheme in terms of computation complexity and communication overhead of SU and UF by deploying it in the real environment.

6.1 Evaluation Environment

In order to measure the integrated performance, we implement the proposed CPSS in smart phones and workstation. Specifically, smart phones with 2.2 GHz eight-core processor, 3 GB RAM, Android6.0 and a workstation with 2.0 GHz six-core processor, 64 GB RAM, Ubuntu are chosen to evaluate SU, UF and SS, respectively, which are connected through 802.11g WLAN. Based on the proposed scheme, we construct a social application and install it on smart phones to evaluate SU and UF, then, we build SS on the workstation. As shown in Fig. 5, SU can register in SS, search her/his friends, and display result in the smartphone. In order to evaluate CPSS in the real environment, the street map in Xi'an is adopted in our application.



(a) m=20 (b) m=40 (c) m=60 (d) m=80

(e) m=100 (f) m=120 (g) m=140 (h) m=160

Fig. 6. Average running time of CPSS in SU vs PDCP.

6.2 Computation Complexity

The proposed PSS algorithm requires mathematical operations with random numbers to protect users' sensitive data from social application servers and attackers. Hence let us quantify the mathematical operations required for the proposed algorithm in SU and UF. Specifically, we assume that the number of search polygon vertexes is n, and SU has m online friends. In the process of blurring the search polygon data, it requires $14n$ multiplication operations. When to generate search result, each UF needs to do $8n$ multiplication operations. After receiving the search results from UF, it will cost $4mn$ multiplication operations for SU to read them. Let us define the time complexity for one multiplication as t_{mul}. Therefore, the total computation complexity of SU and UF are $(14n + 4mn) * t_{mul}$ and $8n * t_{mul}$, respectively.

Our PSS algorithm uses lightweight two-party random masking and polynomial aggregation techniques. Different from other time-consumption homomorphic encryption techniques, it can largely reduce the encryption times for mobile terminals while providing accurate proximity detection results. In the following, for the comparison with CPSS, we select an enhanced proximity detection for convex polygons (PDCP) [26], which adopts the same point in convex polygon strategies as CPSS. Denote that the search domain size is measured by l and the time complexity of exponentiation operation is presented by t_{exp}. Therefore, for PCDP, the computation complexities of SU and UF are $(3n + 2m + 3mn + 4l * mn) * t_{exp} + (8n + 4m + 6mn + l * mn) * t_{mul}$ and $(12n + 4l * n + l^2 * n + 9) * t_{mul} + (4n + 4l * n + 9) * t_{exp}$, respectively.

Table 1. Computation complexity of CPSS and PDCP

	CPSS	PDCP
SU	$(14n + 4mn) * t_{mul}$	$(3n + 2m + 3mn + 4l * mn) * t_{exp} + (8n + 4m + 6mn + l * mn) * t_{mul}$
UF	$8n * t_{mul}$	$(12n + 4l * n + l^2 * n + 9) * t_{mul} + (4n + 4l * n + 9) * t_{exp}$

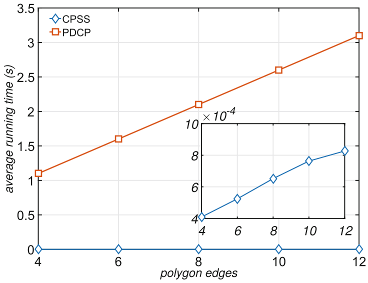
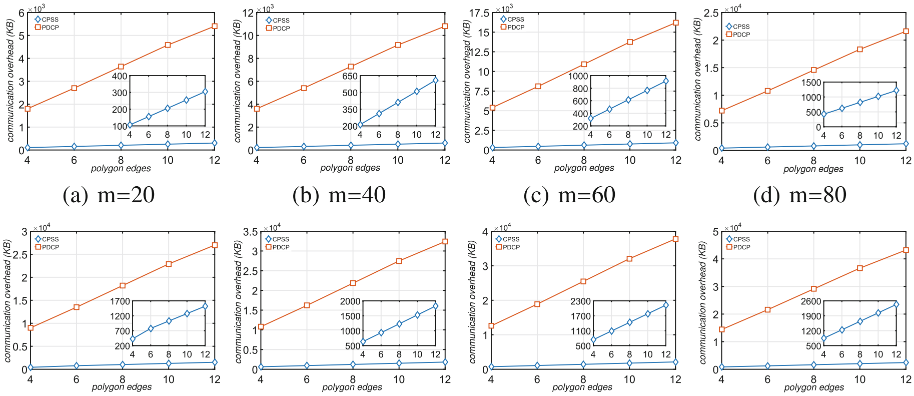


Fig. 7. Average running time of CPSS in UF vs PDCP.

Table 1 presents the computation complexity comparison of CPSS and PDCP. It is obvious that our proposed CPSS can achieve privacy-preserving proximity detection with low computatuon overhead. We test the computation overhead of CPSS and PDCP in SU for various number of SU's friends, and plot the average running time by varying the input number of search polygon edges from 4 to 12 in Fig. 6. It can be obviously seen that with the increase number of polygon edges, the computation overhead of PDCP in SU increase hugely, which is much higher than that of our proposed CPSS. In Fig. 7, 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 PDCP is much higher than that of our proposed CPSS, and increases extremely, which verify the above analysis of computation complexity. In conclusion, our proposed CPSS can achieve better efficiency in terms of computation overhead in SU and UF.



(a) m=20 (b) m=40 (c) m=60 (d) m=80

(e) m=100 (f) m=120 (g) m=140 (h) m=160

Fig. 8. Communication overhead of CPSS vs PDCP.

6.3 Communication Overhead

In order to test the communication overhead, we record the size of search request packet $<\alpha \parallel \beta \parallel Q \parallel SU \parallel TS \parallel MAC_{SU}>$ and result response packet $<R \parallel UF_j \parallel TS \parallel MAC_{UF_j}>$ with different number of polygon edges and SU's friends, and compare with PDCP in one round. As shown in Fig. 8, with the increase of the polygon edges, the communication overhead of PDCP significantly increases and it is much higher than that of our proposed CPSS scheme when the number of SU's friends does not change. Although the communication overhead of our proposed CPSS scheme also increases when the numbers of polygon edges and SU's friends are large, it is still much lower than that of PDCP. In addition, SU needs to interact with UF twice in CPSS, and nine times in PDCP. In conclusion, our proposed CPSS framework can accomplish better efficiency in terms of communication overhead.

7 Conclusion

In this paper, an efficient and privacy-preserving proximity detection scheme with convex polygon spatial search is proposed, which algorithmically improved the privacy-preserving cosine similarity computing protocol and point in convex polygon strategies to achieve efficiency and privacy-preserving. The proposed scheme is based on randomisation technique and only relies on multiplication and addition. In this scheme, LBSAs users can access proximity detection service without divulging their privacy. It is proved that our scheme is secure in security analysis, and extensive experiments show that it is highly efficient in terms of computation complexity and communication overhead.

Availability

The implementation of the proposed CPSS scheme and relevant information can be downloaded at http://xdzhuhui.com/demo/CPSS.

Acknowledgement. H. Zhu is supported in part by National Natural Science Foundation of China (no. 61672411 and U1401251), National Key Research and Development Program of China (no. 2017YFB0802201), Natural Science Basic Research Plan in Shaanxi Province of China (no. 2016JM6007), Research Foundations for the Central Universities of China (no. JB161507), Research Foundations for Science and Technology on Communication Networks Laboratory (no. KX172600023), and China 111 Project (no. B16037).

R. Lu is supported in part by Natural Sciences and Engineering Research (NSERC) Discovery (no. Rgpin 04009), NBIF Start-Up (Nbif Rif 2017-915012), URF (no. Urf Nf-2017-05), and HMF (no. Hmf 2017 Ys-4).

References

1. Valente, T.W.: Network interventions. Science **337**(6090), 49–53 (2012)
2. Zhu, H., Lu, R., Huang, C., Chen, L., Li, H.: An efficient privacy-preserving location based services query scheme in outsourced cloud. IEEE Trans. Veh. Technol. **65**(9), 7729–7739 (2016)
3. Puttaswamy, K.P., Zhao, B.Y.: Preserving privacy in location-based mobile social applications. In: Proceedings of the Eleventh Workshop on Mobile Computing Systems and Applications, pp. 1–6. ACM (2010)
4. Li, K.A., Sohn, T.Y., Huang, S., Griswold, W.G.: Peopletones: a system for the detection and notification of buddy proximity on mobile phones. In: Proceedings of the 6th International Conference on Mobile Systems, Applications, and Services, pp. 160–173. ACM (2008)
5. Bolic, M., Rostamian, M., Djuric, P.M.: Proximity detection with RFID: a step toward the internet of things. IEEE Pervasive Comput. **14**(2), 70–76 (2015)
6. Huang, C., Lu, R., Zhu, H., Shao, J., Alamer, A., Lin, X.: EPPD: efficient and privacy-preserving proximity testing with differential privacy techniques. In: 2016 IEEE International Conference on Communications (ICC), pp. 1–6. IEEE (2016)
7. Chen, Q., Ye, A., Xu, L.: A privacy-preserving proximity detection method in social network. In: Proceedings of the International Conference on Internet of Things and Cloud Computing. ACM (2016). Article No. 68
8. Šikšnys, L., Thomsen, J.R., Šaltenis, S., Yiu, M.L.: Private and flexible proximity detection in mobile social networks. In: 2010 Eleventh International Conference on Mobile Data Management, pp. 75–84. IEEE (2010)
9. Zhu, H., Liu, F., Li, H.: Efficient and privacy-preserving polygons spatial query framework for location-based services. IEEE Internet Things J. **4**(2), 536–545 (2016)
10. Enserink, M.: Risk of exposure. Science **347**(6221), 498–500 (2015)
11. Li, L., Lu, R., Choo, K.K.R., Datta, A., Shao, J.: Privacy-preserving-outsourced association rule mining on vertically partitioned databases. IEEE Trans. Inf. Forensics Secur. **11**(8), 1847–1861 (2016)
12. Peng, J., Meng, Y., Xue, M., Hei, X., Ross, K.W.: Attacks and defenses in location-based social networks: a heuristic number theory approach. In: 2015 International Symposium on Security and Privacy in Social Networks and Big Data (SocialSec), pp. 64–71. IEEE (2015)

13. Huang, C., Yan, Z., Li, N., Wang, M.: Secure pervasive social communications based on trust in a distributed way. IEEE Access **4**, 9225–9238 (2016)
14. Wang, B., Li, M., Wang, H.: Geometric range search on encrypted spatial data. IEEE Trans. Inf. Forensics Secur. **11**(4), 704–719 (2016)
15. Niu, B., Zhu, X., Li, Q., Chen, J., Li, H.: A novel attack to spatial cloaking schemes in location-based services. Future Gener. Comput. Syst. **49**, 125–132 (2015)
16. Ohno-Machado, L.: To share or not to share: that is not the question. Sci. Transl. Med. **4**(165), 165cm15 (2012)
17. Lu, R., Zhu, H., Liu, X., Liu, J.K., Shao, J.: Toward efficient and privacy-preserving computing in big data era. IEEE Netw. **28**(4), 46–50 (2014)
18. Wang, Q., Xu, C., Sun, M.: Multi-dimensional k-anonymity based on mapping for protecting privacy. J. Softw. **6**(10), 1937–1944 (2011)
19. Sharma, V., Shen, C.C.: Evaluation of an entropy-based k-anonymity model for location based services. In: 2015 International Conference on Computing, Networking and Communications (ICNC), pp. 374–378. IEEE (2015)
20. Firoozjaei, M.D., Yu, J., Kim, H.: Privacy preserving nearest neighbor search based on topologies in cellular networks. In: 2015 IEEE 29th International Conference on Advanced Information Networking and Applications Workshops (WAINA), pp. 146–149. IEEE (2015)
21. Chow, C.Y., Mokbel, M.F., Liu, X.: Spatial cloaking for anonymous location-based services in mobile peer-to-peer environments. GeoInformatica **15**(2), 351–380 (2011)
22. Yi, X., Paulet, R., Bertino, E., Varadharajan, V.: Practical approximate k nearest neighbor queries with location and query privacy. IEEE Trans. Knowl. Data Eng. **28**(6), 1546–1559 (2016)
23. Sweeney, L.: k-Anonymity: a model for protecting privacy. Int. J. Uncertainty Fuzziness Knowl. Based Syst. **10**(05), 557–570 (2002)
24. Gedik, B., Liu, L.: Protecting location privacy with personalized k-anonymity: architecture and algorithms. IEEE Trans. Mobile Comput. **7**(1), 1–18 (2008)
25. Wang, S., Wang, X.S.: In-device spatial cloaking for mobile user privacy assisted by the cloud. In: 2010 Eleventh International Conference on Mobile Data Management, pp. 381–386. IEEE (2010)
26. Mu, B., Bakiras, S.: Private proximity detection for convex polygons. In: Proceedings of the 12th International ACM Workshop on Data Engineering for Wireless and Mobile Acess, pp. 36–43. ACM (2013)
27. Hu, P., Mukherjee, T., Valliappan, A., Radziszowski, S.: Homomorphic proximity computation in geosocial networks. In: 2016 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), pp. 616–621. IEEE (2016)
28. Thomas, T.: Secure two-party protocols for point inclusion problem. Int. J. Netw. Secur. **9**(1), 1–7 (2009)
29. Feito, F., Torres, J.C., Urena, A.: Orientation, simplicity, and inclusion test for planar polygons. Comput. Graph. **19**(4), 595–600 (1995)
30. Zhu, H., Liu, X., Lu, R., Li, H.: Efficient and privacy-preserving online medical pre-diagnosis framework using nonlinear SVM. IEEE J. Biomed. Health Inform. **21**(3), 1 (2016)