ISSN No:-2456-2165

A Privacy Preserving Approach in Location Difference Based on Proximity Detection Protocol

1st B.Chandra
Assitant professor
Department of Information Technology
Easwari Engineering College (Anna University)
Chennai, India.

2ndBhavani R
Department of Information Technology
Easwari Engineering college (Anna University)
chennai, India

3rdKiruthika S
Department of Information Technology
Easwari Engineering college(Anna University)
Chennai, India

Abstract:- **Proximity detection is one of the most common location-based application in daily life when user intent to find their friends who get into their proximity. The tremendous growth of the Internet has significantly reduced the cost of obtaining and sharing information about individuals, raising many concerns about user privacy. Spatial queries pose an additional threat to privacy because the location of a query may be sufficient to reveal sensitive information about the querier Studies on protecting user privacy information during the detection process have been widely concerned. Accordingly, a location difference-based proximity detection protocol is proposed based on the Paillier cryptosystem for the purpose of dealing with the above shortcomings. The framework can preserve users 'location privacy in arbitrary local area and can maintain a good utility for both the system and every user. We evaluate our framework thoroughly towards real-world data traces. The results validate that the framework can achieve a good performance.**

Keywords:- *Location privacy, Paillier cryptosystem, privacy preserving, private proximity detecting.*

I . INTRODUCTION

Fog computing also known as fog or fogging, is a decentralized computing infrastructure in which data, compute, storage and applications are distributed in the most logical, efficient place between the data source and the cloud. Fog computing essentially extends cloud computing and services to the edge network, especially in the social network. A Location Difference-based Proximity Detection Protocol is proposed to solve the privacy preserving issue for the proximity detection in a fog computing system. Location Based services(LBS) are increasingly accessed through the mobile devices. This trend forced companies such as Google, Facebook, Apple, and Foursquare to provide services which incorporate location information of users. Today, almost all devices such mobile wireless Phones and tablets have GPS to gather the location information of their users. The major issue in sharing location information is the level of privacy.

II. EXISTING SYSTEM

Our system requires the existence of a social network, i.e., a graph that captures trust relationships between users. Our protocols allow detection of proximity between any two users connected by an edge and we assume the existence of shared secret keys between connected users .The reason we only allow proximity testing between adjacent nodes in a social network is that proximity detection between strangers is a useful functionality, but is impossible to do efficiently and privately in a client-server model.

The reason is that either the server will need to learn some information about users' locations, or it will pair of users identically, resulting in overall bandwidth requirements quadratic in the number of users, unless limited to pairs of friends. revealing even a minimal amount of information about users' locations (e.g., the single-bit outcome of proximity testing between pairs of users) to the server results in an unacceptable privacy leak when aggregated over time and users

III. PROPOSED SYSTEM

The location difference -based proximity detection protocol is able to achieve the data sharing among friends with anti-closure of personal information, especially in the fog computing systems. In the case, the data transmission among non-friends or non-neighbor friends are denied. Yet, the information exchanging between friends is carried out the premise of personal privacy protection. In order to achieve private proximity detection, secure two party homophoric encryption computation was proposed. It resolves the above problem which is called as and a location difference-based proximity protocol.

IV. RELATED WORKS

Panos Kalnis[1]- The increasing trend of embedding positioning capabilities (e.g., GPS) in mobile devices facilitates the widespread use of Location Based Services. For such applications to succeed, privacy and confidentiality are essential. Existing privacy enhancing

techniques rely on encryption to safeguard communication channels, and on pseudonyms to protect user identities.

Nevertheless, the query content may disclose the physical location of the user. In this paper, we present a framework for preventing location based identity inference of users who issue spatial queries to Location Based Services. We propose transformations based on the well-established K-anonymity concept to compute exact answers for range and nearest neighbor search, without revealing the query source. Our methods optimize the entire process of anonym zing the requests and processing the transformed spatial queries. Extensive experimental studies suggest that the proposed techniques are applicable to real-life scenarios with numerous mobile users.

Peter Chapman [2] -Smart phones are becoming some of our most trusted computing devices. People use them to store highly sensitive information including email, passwords, financial accounts, and medical records. These properties make smart phones an essential platform for privacy-preserving applications. To date, this area remains largely unexplored mainly because privacy-preserving computation Protocols were thought to be too heavyweight for practical applications, even for standard desktops. We propose using smart phones to perform secure multi-party computation.

The limitations of smart phones provide a number of challenges for building such applications. In this paper, we introduce the issues that make smart phones a unique platform for secure computation, identify some interesting potential applications, and describe our initial experiences creating privacy-preserving applications on Android devices.

Sergio Mascetti Claudio Bettini Dario Freni DICo[3]- Proximity based services are location based services (LBS) in which the service adaptation depends on the comparison between a given threshold value and the distance between a user and other (possibly moving) entities.

While privacy preservation in LBS has lately received much attention, very limited work has been done on privacy-aware proximity based services. This paper describes the main privacy threats that the usage of these services can lead to, and proposes original privacy preservation techniques offering different trade-offs between quality of service and privacy preservation. The properties of the proposed algorithms are formally proved, and an extensive experimental work illustrates the practicality of the approach.

Leye Wang[4]- In traditional mobile crowd sensing applications, organizers need participants' precise locations for optimal task allocation, e.g., minimizing selected workers' travel distance to task locations. However, the exposure of their locations raises privacy concerns. Especially for those who are not eventually selected for any task, their location privacy is sacriced in vain. Hence, in this paper, we propose a location privacy-preserving task allocation framework with geoobfuscation to protect users' locations during task assignments.

Specially, we make participants obfuscate their reported locations under the guarantee of deferential privacy, which can provide privacy protection regardless of adversaries' prior knowledge and without the involvement of any third-part entity. In order to achieve optimal task allocation with such differential geo-obfuscation, we formulate a mixed-integer non-linear programming problem to minimize the expected travel distance of the selected workers under the constraint of differential privacy. Evaluation results on both simulation and real-world user mobility traces show the effectiveness of our proposed framework. Particularly, our framework outperforms Laplace obfuscation, a state-of the art differential geo-obfuscation mechanism, by achieving 45% less average travel distance on the real-world data.

V. METHODOLOGIES

A. System architecture

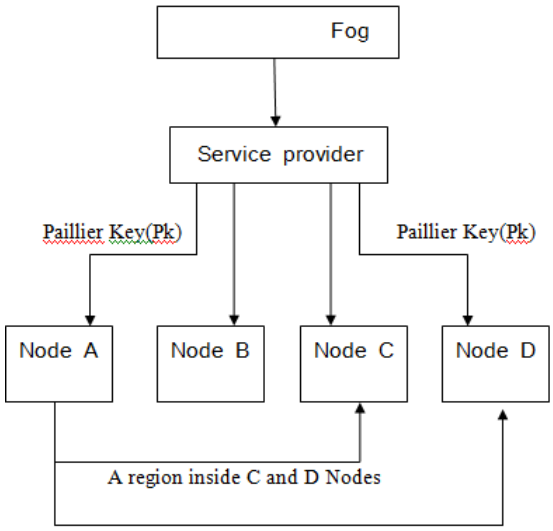

Fig 1:- System architecture

B. Network Formation

In this module, Initially Service provider will generate first with three components like Paillier key, view nodes region, Encryption. And then we have to create nodes under the wireless network with latitude and longitude. Each and every node contains proximity region in this network. we consider a multi-hop WN consisting of a number of fog nodes. Using multicast socket, all nodes are used to detect the neighbor nodes. The fog node maintained neighbors list it is used to find all possible path to reach destination. And it contains the private key and public key.once enter the network it will automatically create polygon proximity region using latitude and longitude

C. Paillier key distribution

In this module, after enter the node in network each and every node receive the paillier key from service provider. Every nodes one by one get the key using with

private key. And then node A (represented a task initiator or the host fog node), her friend B and a fog sever SP. We assume that both A and B should have mobile devices with GPS and basic communication capabilities, If A want to add a friend in this network ,Select the friend name and then give a friend request after accept that person your are friends in this network.

In the scenario that your friends get into your vicinity, a Service Provider (SP) will remind you based on your demand that the friend is close to you. For example, when A wants to know which of her friends are in the same park with her, she will consider the park as her vicinity region and send a query command to the SP to find her friends within the same park.

D. Proximity detection

In this module , Alice wants to know which of her friends are in the same park .A provide a latitude and longitude to service provider and then service provider encrypt the latitude and longitude with A node location. Again send to node A. After receiving the details node A broadcast this details with her friends. B and other nodes are receive the encrypted message using with paillier key and then nodes are return our current location to send service provider then service provide will check the each and every nodes in within A proximity region or not ,using with proximity detection techniques.

- *Data Communication*

In this module after the detection techniques A got a near by friend in your emergency list. If you want to communicate with your friend first select and send message to destination and you cannot communicate with other friend like that not your proximity.

VI. PRELIMINARIES

In this section, we first introduce the system model and the problem formulation for the PPD, then briefly describethe Paillier cryptosystem to achieve privacy persevering in the data transmission process.

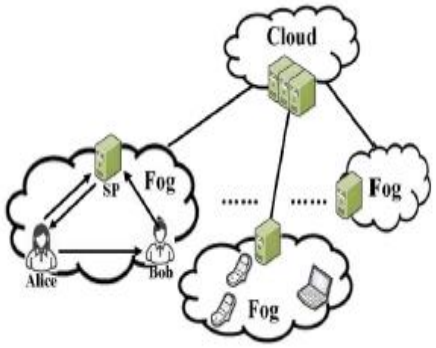

Fig 2:- Preliminaries

A. System model

Under the cloud network, our system model, shown in Fig 2., consists of three types of entities of fog network, also called as the fog nodes, including Alice (represented a task initiator or the host fog node), her friend Bob and a fog sever.

SP. We assume that both Alice and Bob should have mobile devices with GPS and basic communication capabilities, so as to allow them to determine the actual location and communicate.

With other entities. The goal of this paper is not only to achieve proximity detection, but also to protect the location privacy of fog nodes. Therefore, it is essential to define the privacy threats and requirements in details as below.

According to Fig. 2., we assume that Alice can specify an arbitrary polygon P as a vicinity region including her position, which consists of n vertices $\{P0, P1, \ldots, Pn-1\}$. In that case, she also can initiate a query to the local SP to inquire whether Bob is in or on the boundary of the proximity P. To illustrate the privacy preserving in the process of proximity detection, we provide the relevant threats and requirements here

- *Privacy threats*

In our protocol, all the network entities, including the three parts of the system model and other external entities, are treated as potential adversaries. Alice would spare no efforts to get the exact location of Bob, while Bob would try to acquire the location of Alice as well. Meanwhile, the local SP would also try to derive any information of Alice's proximity P and the location of Bob. Moreover, the external malicious attackers would go all out to pick up the exact location information about Alice and Bob. Finally, each party in Fig.2. is assumed as a semi-honest secure model, which means that Alice, Bob and the SP are not collusion with each other. Thus, all the messages will be dealt with in the PPD process.

- *Privacy Requirements*

The privacy requirements of our protocol are listed as follows: first, Alice has rights to launch a query to inquire whether Bob locates in the proximity and only gets a response with FALSE or TRUE for the purpose of keeping Bob's exact location secrecy. Second, Bob cannot access the result of Alice's query or pick up any information related to Alice's location including neither the shape of proximity region P nor the exact location. Third, the local SP should not deduce the exact locations of Alice and Bob. To satisfy the requirements of privacy, we will introduce a homomorphic algorithm named Paillier cryptosystem in Section B.

- *Accuracy Requirement*

In our protocol, we set χ determined by Alice to be the accuracy requirement, which is related to the number of decimal places of the GPS. Generally speaking, our civilian GPS on the user's smart mobile can Be accurate to the seventh places after the decimal point for collecting user's location. It is the reason that the value of χ is set up from 2

to 7. The bigger of χ, the higher precision of the PPD result. For example, when Alice set $\chi = 6$, the whole system deviation can be accurate to 1 m.

B. Homomorphic Encryption

Homomorphic encryption [29] allows certain computation over encrypted data. Paillier cryptosystem [30] is a popular Homomorphic encryption scheme that provides fast encryption and decryption [30], [31], which is a probabilistic asymmetric algorithm based on the decisional composite residuosity problem. It is adopted by the secure scalar product [32], which has been widely used in privacy preserving data mining. It also has been applied to privacy-preserving localization [33] and privacy-preserving biometric identification [34]. The Paillier crypto system is briefly introduced as follows.

- *Key Generation*

An entity selects two large primes p and$q$ and computes $N = p \cdot q$ and $\lambda = lcm(p - 1, q - 1)$,where *lcm* stands for the least common multiple. It then chooses an nonzero integer g such that $gcd(L(g^{\lambda} \bmod N2),N) = 1$, where *gcd* stands for the greatest common divisor, $g \in Z * N$, and $L(x) = [(x - 1)/N]$. The public key and private key are, respectively,$\{N, g\}$ and $\{\lambda\}$.

- *Encryption*

Let $m \in Z^*N$ be a plaintext and $r \in Z^*N$ be a random number. The cipher text of m is computed by

$$E(m) = g^m \cdot r^N \bmod N^2 \quad (1)$$

where $E(\cdot)$ denotes the encryption operation using public key $\{N, g\}$.

- *Decryption*

For the ciphertext$E(m)$, the corresponding plaintext can be computed by

$$\frac{D(E(m)) = L\_E(m)\lambda modN2\_}{L\_g\lambda modN2} mod\ N$$

where $D(\cdot)$ denotes the decryption operation using private key $\{\lambda\}$.

- *Homomorphic*

The Paillier cryptosystem is additively homomorphic as it satisfies the following conditions: given

$\{m1,m2\} \in Z^*N$, we have

$$E(m1) \cdot E(m2) = E(m1 + m2). \quad (3)$$

Furthermore, given $E(m)$ and a constant K, $E(K \cdot m)$ can be computed by

$$E(K \cdot m) = E(m)K. \quad (4)$$

Obviously, the computational cost will grow exponentially with the increase of K. Therefore, the cost during the computation phase of $E(K \cdot m)$ on the smart phone should be great when we select a n-bit integer K for the secure communication.

To reduce the computation, (4) can be represented as

$$E(K \cdot x) = E(\sum k^i.10^i.m) = \prod E(10^i.m)^{ki} \quad (5)$$

where Ki is the value of the ith place of the big integer K. In other words, k can be represented as

$$K = \sum Ki \cdot 10i.$$

Accordingly, we can easily simplify the encryption computation shown in Section A.

VII. PROXIMITY INFORMATION SECURITY EXTRACTION

Considering the individual privacy, an analysis of Paillier based relative location is presented to determine that Bob locates on which side of boundary line of Alice's proximity region. Accordingly, we generate the decision-tree to illustrateour proximity detection process.

A. Paillier-Based Relative Location Analysis

Assuming a line l in fig.3 is an edge of Alice's proximity, we can choose two points $A(xa, ya)$ and $B(xb, yb)$ on the line. Then we analyze Bob's location on the basis of l's slope.

- *Slope of l Is Real Number:* In this case, the slope andintercept of l can be expressed as

$$\frac{k = ya - yb}{xa - xb}$$

and

$$\frac{Rl = xbya - xayb}{xb - xa}$$

SuPpose Bob's exact location is point $Q(xq, yq)$, we can draw a new line lq through Q and paralleling to l, whose slope and intercept are as

$$\frac{k = ya - yb}{xa - xb}$$

and

$$Rq = yq - ya - ybxa - xbxq.$$

Therefore, we now deduce the intercept difference betweenthe two lines, which is

$$\alpha = Rl - Rq = xbya - xaybxb - xa - yq + ya - yb$$
$$xa - xbxq = (xbya - xayb) + (yb - ya)xq + (xa - xb)yqxb - xa = z1 + z2 + z3xb - xa = \beta xb - xa$$

where β is the numerator of α, and $z1$, $z2$, $z3$ is defined as x by $a - xayb$, $(yb - ya)xq$, $(xa - xb)yq$, respectively.

Assume without loss of generality that $xb > xa$, the cipher text of β based on the Paillier encryption system described in Section III-B is

$$E(\beta) = E(z1) \cdot E(z2) \cdot E(z3)$$
$$= E(xbya - xayb)E(yb - ya)xqE(xa - xb)yq \ . \quad (7)$$

- *Slope of l Is Infinite:* As of the equal *x*-coordinates of *A* and *B* (*xa* = *xb*), the intercept difference between the *l* and *lq*

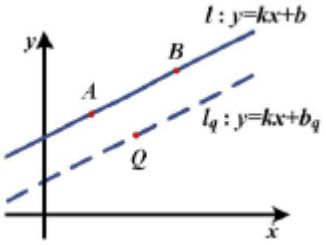

Fig 3:- Relationship between two lines.

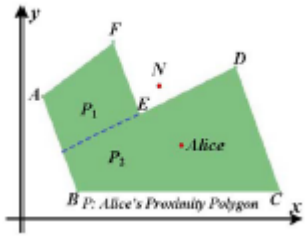

Fig 4:- Alice's proximity.

is represented as $\alpha = Rl - Rq = xa - xq$. Here, we assume that $ya > yb$ for the convenient of discussion. Similar to (7), the value of β in the case of infinite slope of l can be rewritten as

$\beta = \_xa - xq\_(ya - yb)$
$= (xbya - xayb) + (yb - ya)xq + (xa - xb)yq.$ (8)

Therefore, we can also encrypt β, the sign of relative location between two lines, by the Paillier cryptosystem in this case. Then the local SP can compare the encrypted β of the two lines, $E(\beta)$, regardless of the value of l's slope. Specifically, $E(z1)$, $E(z2)$, and $E(z3)$ in (7) should be successively calculated. Obviously, $z1$ can be encrypted easily by the public key, while the followed two parts in (7) have to be computed by (5) for the purpose of simplifying the computation cost. Thus, the encrypted $z2$ and $z3$ can be rewritten as

$E(z2) = E\_10n(yb - ya)\_xn \cdot \cdot \cdot E(yb - ya)x0$
$E(z3) = E\_10n(xa - xb)\_yn \cdot \cdot \cdot E(xa - xb)y09)$
where $xq = \_ni$
$=0\ xi \cdot 10i$, $yq = \_nj$
$=0\ yj \cdot 10i$.

After that, the local SP will deduce the relative location of the two lines by decrypting the value of $E(\beta)$, so that we can determine Bob locates on which side of the edge *l*. In this way, the final result of the proximity detection will be determined when every edge of Alice's proximity is tested. In order to further simplify the computation process and improve the algorithm efficiency, we exploit the decision-tree theory to detect the Alice's proximity, which will be discussed in detail in the following section.

B. Generation of Decision Tree

To illustrate how to infer the relationship between Bob's location and any edge of a polygon specified by Alice when considering privacy preserving, we first show a diagram, Fig. 5, which shows the Alice's proximity polygon *P* and her friend Bob whose location is *Q(xq, yq)*. Before generate the decision tree for the purpose of proximity detection, some definitions are presented as follows.

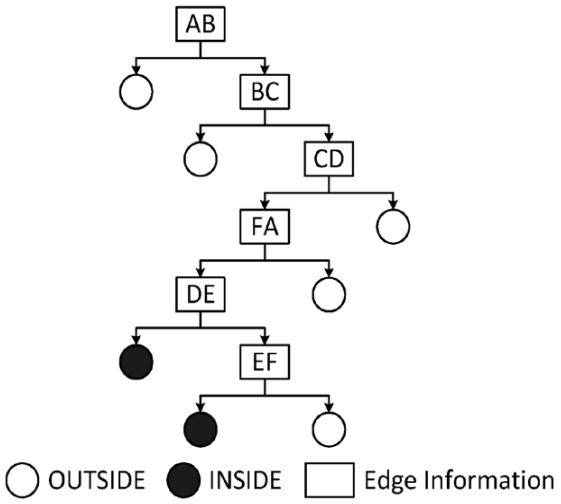

Fig 5:- Decision-tree.

- *Definition 1 (Convex Edge and Concave Edge)*

If all of Alice's proximity region is located on a single side of an edge and its extension line, we define this edge as a convex edge. Otherwise we define the edge as a concave edge. In Fig. 3, *AB*, *BC*, *CD*, and *FA* are defined as convex edges because all of Alice's proximity area is located on the same side of these edges. On the contrary, *DE* and *EF* are the concave edges.

- *Definition 2 (Upper Side and Lower Side):*

Recall the definition of intercept in last section, we define that Bob is located on the upper side of the edge if $\beta < 0$, otherwise Bob is located on the lower side. Take Fig. 3 as an example, *N* locates on the upper side of line *AB* and on the lower side of line *FA*.

- *Definition 3 (INSIDE and OUTSIDE)*

We define Bob is INSIDE of Alice's proximity if Bob locates within the Alice's proximity area *P*. Otherwise, Bob is OUTSIDE of *P*. Fig. 4 demonstrates an example of proximity detection using decision-tree theory for Fig. 3. Each edge of Alice's proximity is a decision node in the tree, which has two child nodes to represent the lower or the upper side of the edge. Particularly, the left child node means the lower side of the edge, whose index of the sub tree is defined as $\gamma = 1$, whereas $\gamma = -1$ indicates the right node that is located on the upper side of the edge. Besides the decision nodes explained above, we also employ.

Definition 3 to end the query process of proximity detection for Bob. In o+ other words, we can finish the detection process and draw the final conclusion that whether

ISSN No:-2456-2165

Bob is inside Alice's proximity when a detected edge is in the INSIDE state.

Generally, Alice's proximity polygon may contain both convex and concave edges. For convenience in this paper, we will usually first create the decision-tree by Alice's convex edges and followed by the concave edges. In other words, all convex edges in Fig. 3, which are *AB*, *BC*, *CD*, and *FA*, should be processed first. Obviously, the proximity area is located on the upper side of the line *AB*, which points to the next convex edge *BC* on the right child node of the sub tree. On the contrary, the left node of the sub tree labeled "OUTSIDE" means the ending detection. It is the same as the other convex edges, including *BC*, *CD*, and *FA*. Note that the last convex edge should point to the first concave edge according to Bob's relative location.

Unlike the convex edge analysis, we should exploit extension line of the concave edge to divide Alice's proximity. Take the extension line of *DE* in Fig. 3 as an example. It may divide

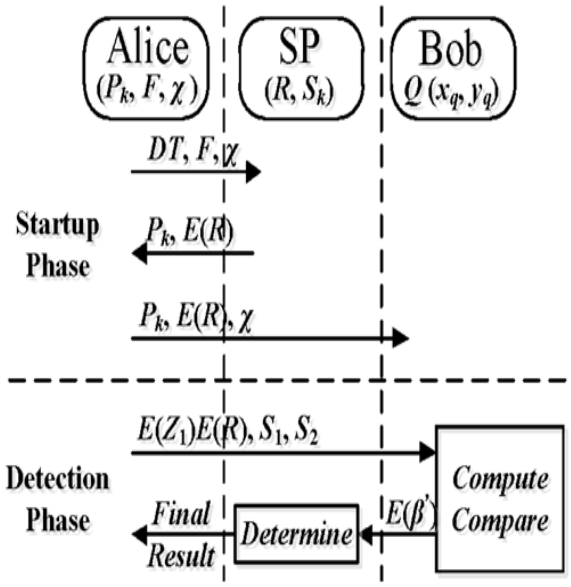

Fig 6:- Detailed Lo DPD.

Algorithm 1 System Startup

- *Require*

Friends set *F*; Proximity Polygon *P* with *N* edges;

1: Alice forms and sends the encrypted messages of *DT* andχ to the local SP.

2: SP generates the Paillier key (*Pk*, *Sk*), selects a big integer *R*, and sends *Pk* and *E(R)* to Alice.

3: Alice broadcasts *Pk* and *E(R)* to all of her friends. the proximity region into two sub-areas, which are the upperpart *P*1 and the lower part *P*2. Obviously, there was no doubting that Bob locates in Alice's proximity if he locates in the part *P*2 that is on the lower side of *DE*. So that we assign the "INSIDE" label to the left child node and the next convexedge *EF* to the right child node. Similarly, if Bob locates on the lower side of *EF*, INSIDE and OUTSIDE are labeled to the left and right child nodes, respectively.

VIII. CONCLUSION

A Location Difference-based Proximity Detection Protocol is proposed to solve the privacy preserving issue for the proximity detection in a fog computing system, which exploit the Paillier encryption algorithm and the decision-tree theory. Without the collusion scenario, we define a difference that is used to determine the relative location between a edge and Bob's location in the protocol for the purpose of ensuring privacy.

During the detection, the parameters are transmitted among Alice, Bob and the SP in the Paillier encryption form to keep out of the external malicious attacks. Analyses and simulation results clearly explain that our protocol outperforms the traditional PPD method in both communication cost and CPU cost

IX. ACKNOWLEDGEMENT

The authors would like to thank all reviewers who have helped improve the quality of this paper.

REFERENCES

[1]. Z. Cai, Z. He, X. Guan, and Y. Li, "Collective data-sanitization forpreventing sensitive information inference attacks in social networks,"
[2]. Y. Wang et al., "An incentive mechanism with privacy protection inmobile crowdsourcing systems," Comput. Netw., vol. 102, pp. 157–171,Jun. 2016.
[3]. I. Stojmenovic, "Fog computing: A cloud to the ground supportfor smart things and machine-to-machine networks," in Proc. Aust.Telecommun. Netw. Appl. Conf. (ATNAC), Southbank, VIC, Australia,2014, pp. 117–122.
[4]. J. Wang et al., "Differentially private k-anonymity: Achieving queryprivacy in location-based services," in Proc. Int. Conf. Identification Inf.Knowl. Internet Things (IIKI), Beijing, China, Oct. 2016, pp. 1–6.
[5]. Z. He et al., "An energy efficient privacy-preserving content sharingscheme in mobile social networks," Pers. Ubiquitous Comput., vol. 20,no. 5, pp. 833–846, 2016.
[6]. A. Stefanidis, A. Crooks, and J. Radzikowski, "Harvesting ambientgeospatial information from social media feeds," GeoJ., vol. 78, no. 2,pp. 319–338, 2013.
[7]. P. F. Riley, "The tolls of privacy: An underestimated roadblock for electronictoll collection usage," Comput. Law Security Rev., vol. 24, no. 6,pp. 521–528, 2008.
[8]. J. Y. Tsai, P. G. Kelley, L. F. Cranor, and N. Sadeh, "Location-sharingtechnologies: Privacy risks and controls," J. Law Policy Inf. Soc., vol. 6, no. 2, p. 119–151, 2010.

[9]. X. Zheng, Z. Cai, J. Li, and H. Gao, "Location-privacy-aware reviewpublication mechanism for local business service systems," in Proc. 36th Annu. IEEE Int. Conf. Comput. Commun. (INFOCOM), Atlanta, GA,USA, May 2017, pp. 1–9.

[10]. X. Lin, H. Hu, H. P. Li, J. Xu, and B. Choi, "Private proximity detectionand monitoring with vicinity regions," in Proc. 12th Int. ACM Workshop Data Eng. Wireless Mobile Acess, New York, NY, USA, 2013, pp. 5–12.

[11]. S. Mascetti, C. Bettini, D. Freni, and X. S. Wang, "Spatial generalisationalgorithms for LBS privacy preservation," J. Location Based Services, vol. 1, no. 3, pp. 179–207, 2007.

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

[13]. Z. He, Z. Cai, Y. Sun, Y. Li, and X. Cheng, "Customizedprivacy preserving for inherent data and latent data," Pers. UbiquitousComput., vol. 21, no. 1, pp. 43–54, 2017.

[14]. L. Zhang, Z. Cai, and X. Wang, "FakeMask: A novel privacy preservingapproach for smartphones," IEEE Trans. Netw. Service Manag., vol. 13,no. 2, pp. 335–348, Jun. 2016.

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

[16]. L. Zhang, X. Wang, J. Lu, P. Li, and Z. Cai, "An efficient privacypreserving data aggregation approach for mobile sensing," SecurityCommun. Netw., vol.9, no. 16, pp. 3844–3853, 2016.

[17]. M. Gruteser and D. Grunwald, "Anonymous usage of location-basedservices through spatial and temporal cloaking," in Proc. 1st Int. Conf. Mobile Syst. Appl. Services, San Francisco, CA, USA, 2003, pp. 31–42.

[18]. P. Golle and K. Partridge, "On the anonymity of home/work locationpairs," in Proc. Int. Conf. Pervasive Comput., Nara, Japan, 2009,pp. 390–397.

[19]. A. Khoshgozaran and C. Shahabi, "Blind evaluation of nearest neighbour queries using space transformation to preserve location privacy," in Proc. Int. Symp. Spatial Temporal Databases, Boston, MA, USA, 2007,pp. 239–257.