I Know Where You Are: Thwarting Privacy Protection in Location-Based Social Discovery Services

Minhui Xue*‡, Yong Liu†, Keith W. Ross†‡, Haifeng Qian*
*Department of Computer Science, East China Normal University, Shanghai, China
†Department of Computer Science and Engineering, New York University, New York, USA
‡NYU Shanghai, Shanghai, China
{minhuixue, yongliu, keithwross}@nyu.edu, hfqian@cs.ecnu.edu.cn

***Abstract*—Location-based Social Discovery (LBSD) services enable users to discover their geographic neighborhoods to make new friends. Original LBSD services were designed to provide the exact distances to nearby users. It has been shown that it is easy to pinpoint any target user's location by using trilateration based on the exact distances from three fake GPS locations to the target user. To thwart the trilateration attack, contemporary LBSD services then began to report distances of nearby users in concentric bands, e.g., bands of 100 meters. In this paper, we investigate the user location privacy leakage problem in LBSD services reporting distances in discrete bands. Using number theory, we analytically show that by strategically placing multiple virtual probes with contrived fake GPS locations, one can nevertheless pinpoint user locations in band-based LBSD. Our methodology guarantees to pinpoint any reported user within an area bounded by one square meter, even for LBSD services using large bands (such as $100m$ as used by WeChat). To the best of our knowledge, this is the first work that explicitly exploits and quantifies user location privacy leakage in band-based LBSD services. Our study is expected to draw more public attention to this serious privacy issue and hopefully motivate better privacy-preserving LBSD designs.**

I. Introduction

The skyrocketing growth of location-based social networks (LBSNs) has gained billions of users driven by the wide proliferation of both smartphone technology and ubiquitous location-based services (LBSs). A popular type of LBS, termed Location-based Social Discovery (LBSD) services, provides a smartphone user a list of nearby people (not necessarily friends) along some indication of how far away the users are. The smartphone user can then exchange messages with the discovered nearby users, thereby attempting to make new friends or meet up with existing nearby friends.

In real-world LBSD services, there has been a plethora of prevalent applications including WeChat, Momo, and Skout. WeChat – which provides other services in addition to LBSD – boasts more than 600 million users and is currently used by essentially all smartphone users in China on a daily basis. In order to protect users' privacy, these services generally do not report the exact longitude and latitude locations of the nearby users. In the first generation, they instead reported to the smartphone user exactly how far away each of the nearby users are. For example, suppose Alice is a smartphone and runs the LBSD application. The first generation applications would report to Alice information like "Bob is 176 meters away and Clark is 227 meters away". This information would not provide Bob's and Clark's exact locations but instead locate them to circles of radius 176 and 227 meters, respectively. Unfortunately, as we review in Section II, it has been shown that it is easy to pinpoint any reported user's location by using trilateration based on the exact distances from three fake GPS locations to the reported user. (Fake GPS is an App that lets any smartphone user – such as Alice – to configure her fake longitude and latitude locations to any place in the world.)

To thwart the trilateration attack, contemporary LBSD services then began to report distances of nearby users in concentric bands, e.g., they might say "Bob is between 100 and 200 meters away, and Clark is within 200 and 300 meters away". This would put Bob in a large circular band of $\pi 200^2$ - $\pi 100^2$ m^2 and Clark in an even larger circular band of $\pi 300^2$ - $\pi 200^2$ m^2. By reporting distances in bands, one can no longer directly apply trilateration to pinpoint the locations of the discovered users. WeChat, for example, currently uses this concentric band approach when reporting distances. This band-based approach therefore seemingly protects users' privacy to a much greater extent.

We show, nevertheless, that by using fake GPS to carefully place multiple virtual probes, we can nevertheless pinpoint the discovered users' locations, even when band-based privacy protection mechanisms are used. Our methodology guarantees to pinpoint any reported user within an area bounded by one square meter, even for LBSD services using large bands (such as $100m$ as used by WeChat). To the best of our knowledge, this is the first work that explicitly exploits and quantifies user location privacy leakage in band-based LBSD services.

In this work, we adopt a generic approach to prove that it is possible for an ordinary user of an LBSD network to pinpoint individuals' locations and trace their mobility patterns in any targeted area. The attack not only can target a specific individual, but from any one geographical location (such as Washington D.C.), it can also target any other geographical region (e.g., Beijing) and monitor all the users using the service in that region. Obviously, if a weak adversary can monitor a region in a city, then so can a stronger adversary such as a government intelligence agency. We emphasize, however, that a person can only be discovered if he is a user of the LBSD service. For example, if a WeChat user only uses WeChat for messaging friends and posting photos, and never uses WeChat's LBSD service, then the user is not locatable by the methods described in this paper. And a user's mobility is only traceable if the user repeatedly uses the LBSD service (repeatedly queries "*People Nearby*" in WeChat).

In this paper, we first consider a one-dimensional version of the problem. We employ number theory to prove that under some easily satisfiable conditions, we can locate a reported user to within a half meter. We then extend the methodology to the two-dimensional approach by placing virtual probes as a lattice of equidistant points (honeycomb). Our approach combines the distance observed by several probes to determine an overlapping area bounded by $\frac{\sqrt{3}}{2}m^2$ containing the target user. Therefore, we can use such information to pinpoint the users, and even identify their mobility patterns if they repeatedly use the LBSD service. Our analysis shows that current band-based LBSD services fail to protect users' location privacy.

II. Problem Statement

In this section, we first overview the state-of-the-art of LBSD Services and the trilateration attack on exact distance. We then present the problem when LBSD services use band-based distance.

A. LBSD Services and Trilateration Attack

LBSD applications enable a user to find nearby users. In the example of WeChat, which has drawn 600 million users globally since the service was released in January 2011, it provides a "*People Nearby*" LBSD service, which reads in the current geo-localization of the mobile device and returns a list of other WeChat users in geographic proximity, establishing on-the-spot connection among nearby users. One option is to report the exact distance to the nearby users. As reported in [1], LBSD services with exact distance are vulnerable to the so called *Trilateration Attack*. In Euclidean geometry, trilateration is the process of determining relative locations of points by measurement of exact distances, using the geometry of circles. To perform the attack, when a target user lies on three circles from known locations, then the centers of the three circles with their exact radii provide sufficient information to pinpoint the possible locations down to one unique location of the target user.

LBSD applications read in a user's location from his device as long as the user uses the LBSD feature: if a user never triggers the LBSD feature, one's location privacy will not be threatened by the attack. However, Terence Chen *et al.* [2] has observed that only about 6% of the total number of users in their dataset has their locations hidden from public access.

To defend against trilateration attack, contemporary LBSD applications, such as WeChat, Momo, and Skout, adopt obfuscation techniques to blur the location information by having the user's smartphone submit ROUNDUP relative distance instead of the exact relative distance. For example, WeChat reports the relative distance in bands of $100m$. When WeChat shows that the target user is $500m$ away from a certain user, it means that the target user is located in a band area centered at the current user's location with radius r ranging from $400m$ to $500m$. We outline the distance report accuracy and coverage of some prevalent LBSD applications in Table I [3]. In general, we assume that LBSD applications provide the relative distance in bands of K meters. Then the relation between the reported relative distance ω_d and the exact relative distance d can be formalized as follows:

$$\omega_d = \left(\left\lfloor \frac{d}{K} \right\rfloor + 1\right) \times K.$$

TABLE I: Location-based Social Discovery Applications

App	Accuracy Limit	Coverage Limit
WeChat	100m	1km
Momo	10m	N/A
Skout	0.5mile	N/A
Whoshere	100m	N/A
Topface	100m	N/A
SayHi	10m	1000km
iAround	10m	N/A
U+	10m	N/A
LOVOO	100m	27.8km
KKtalk	10m	N/A

B. Adversary Model

Simple trilateration attack no longer works with band distance. In this paper, we develop a new location privacy attack on LBSD applications reporting band distance. In the adversary model, an attacker places multiple virtual probes in a remote geographical region (e.g., pinpoint NYC from Shanghai), which can be done easily by spoofing the LBSD applications with fake GPS locations. Each virtual probe collects nearby LBSD users with the corresponding relative distance bands to the probe. While virtual probes can be deployed in arbitrary locations, we assume that virtual probes are positioned to form a lattice of equidistant points. The attacker is to pinpoint the target user in a lattice by using relative band distances provided by virtual probes based on revised trilateration methodology. We analytically show that by carefully placing multiple virtual probes with contrived fake GPS locations, one can be located somewhere in the highlighted area with three circular rings as illustrated in Figure 1.

More concisely, we consider

- a geographical region as a lattice of equidistant virtual probes, the distance between one probe and any adjacent probe is x;
- each probe reports the relative distance to the target user in bands/units of K;
- x is chosen such that $\gcd(x, K) = 1$, where $x, K \in \mathbb{Z}$ and $x, K \geq 1$.

The problem under study is illustrated in Figure 1.

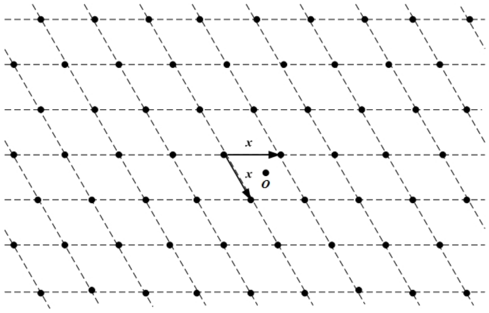

Fig. 1: Lattice (honeycomb)

III. Preliminary Number Theory

In this section, we introduce notations from number theory and review several lemmas which will be used later. Due to space limitations, we only present a snapshot of all the lemmas and theorems we obtained throughout this paper.

Division: Consider two integers $a, b \in \mathbb{Z} = \{..., -1, 0, 1, 2, ...\}$, we say that b divides a iff $\exists\ c \in \mathbb{Z}$, such that $a/b = c$, and write $b \mid a$. **Common Divisor:** For $a, b \in \mathbb{Z}$, we call $d \in \mathbb{Z}$ a common divisor of a and b if $d \mid a$ and $d \mid b$; moreover, we call such a d a **greatest common divisor** of a and b if d is non-negative and all other common divisors of a and b divide d, and write $\gcd(a, b) = d$. **Congruence:** For a positive integer n and for $a, b \in \mathbb{Z}$, we say that a is congruent to b modulo n if $n \mid (a - b)$, and write $a \equiv b \pmod{n}$.

Lemma 3.1: For $a, b \in \mathbb{Z}$, there exist $s, t \in \mathbb{Z}$ such that $as + bt = 1$ if and only if $\gcd(a, b) = 1$.

For simplicity, by using **Extended Euclidean Algorithm** to compute s and t where $|s| \leq |b|$ and $|t| \leq |a|$, the proof of Lemma 3.1 is trivial [4].

Lemma 3.2: For any $y \in \mathbb{Z}$ such that $1 \leq y \leq K - 1$ and any positive integer x such that $\gcd(x, K) = 1$, there exists $n \in \mathbb{Z}$ where $1 \leq n \leq K - 1$, such that

$$n \cdot x \equiv y \pmod{K}.$$

Lemma 3.3: For any $y \in \mathbb{Z}$ such that $1 \leq y \leq K - 1$ and any positive integer x such that $\gcd(x, K) = 1$, there exists a unique $n \in \mathbb{Z}$ where $1 \leq n \leq K - 1$, such that

$$n \cdot x \equiv y \pmod{K}.$$

Remark 3.4: By applying Lemma 3.2 and Lemma 3.3, we can obtain that $f_x(N) \equiv N \cdot x \pmod{K}$ is a permutation over $\mathbb{Z}_K$. For the above s, note that $f_s(N) \equiv N \cdot s \pmod{K}$ is the inversion of $f_x(N)$ over $\mathbb{Z}_K$, i.e., $f_s(f_x(N)) \equiv N \pmod{K}$.

IV. One-Dimensional Adversarial Methodology

In this section, we consider a special One-Dimensional (1-D) case of the problem that we try to detect the target user's location along a line, which is composed of a set of evenly spaced probes. We prove that the accuracy of 1-D case is bounded by half meter.

A. Formulation

We summarize the notations introduced throughout this section in Table II.

TABLE II: Summary of Notations

Symbol	Meaning
O	the target point (the location of the user)
d_{p_i}	the accurate distance between the probe p_i and the target point O
ω_{p_i}	the reported distance between the probe p_i and the target point O
r_{p_i}	the remainder of the distance to probe p_i, i.e., $r_{p_i} \equiv d_{p_i} \pmod{K}$

As illustrated in Figure 2, in the 1-D case, all the virtual probes are evenly spaced along a line. Then we have

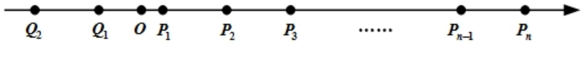

Fig. 2: One-dimensional Line

$$d_{p_i} = d_{p_{i-1}} + x = d_{p_1} + (i-1) \cdot x, \quad \text{where} \quad i \geq 1, \quad \text{(IV.1)}$$

$$x = \left\lfloor \frac{x}{K} \right\rfloor \times K + r_x, \quad \text{where} \quad 0 \leq r_x < K, \quad r_x \in \mathbb{Z},$$

$$\omega_{p_i} = \left(\left\lfloor \frac{d_{p_i}}{K} \right\rfloor + 1 \right) \times K, \quad \text{where} \quad \forall i \in \mathbb{N}.$$

Algorithm 1 takes in x, $\{\omega_{p_i}\}$ as input, and returns the accurate distance to the first probe d_{p_1} after certain iterations. By using Binary Search, the iterative algorithm can be further improved to reduce the computation time. The complexity of Binary Search Method and Simple Iterative Method are also numerically evaluated in Section VI.

Algorithm 1 Iterative Algorithm for 1-D Case

```
Initialization
INPUT
The distance between one probe and any adjacent probe: $x$;
The set of samples for the reported distance: $\{\omega_{p_i}\}$;
Extended Euclidean Algorithm
Compute $s$ such that $s \cdot x + t \cdot K = 1$
$T \leftarrow K$
REPEAT
$T \leftarrow T - 1$
$N = f_s(T) + 1 = T \cdot s \pmod{K} + 1$
    $\Delta(T) = \begin{cases} 1, & \frac{\omega_{p_N}}{K} = \frac{\omega_{p_1}}{K} + \left\lfloor \frac{f_s(T) \cdot x}{K} \right\rfloor; \\ 0, & \text{Otherwise.} \end{cases}$
UNTIL $\Delta(T) = 1$ or $T = 1$
RETURN $D_{p_1} = \omega_{p_1} - T - \frac{1}{2}$
```

Note that the key role of the Algorithm 1 is the equation

$$\frac{\omega_{p_N}}{K} = \frac{\omega_{p_1}}{K} + \left\lfloor \frac{f_s(T) \cdot x}{K} \right\rfloor. \quad \text{(IV.2)}$$

We can use equation IV.2 as a test function to determine the location. Since

$$1 \leq T \leq K - 1, \quad f_s(T) \cdot x \equiv f_x(f_s(T)) \equiv T \pmod{K}.$$

We have

$$f_s(T) \cdot x = \left(\left\lfloor \frac{T \cdot s \cdot x}{K} \right\rfloor - \left\lfloor \frac{T \cdot s}{K} \right\rfloor \cdot x \right) \times K + T. \quad \text{(IV.3)}$$

Thus let $N = f_s(T) + 1$, it follows

$$\frac{\omega_{p_N}}{K} = \left\lfloor \frac{\lfloor \frac{d_{p_1}}{K} \rfloor \times K + r_{p_1} + \lfloor \frac{f_s(T) \cdot x}{K} \rfloor \times K + T}{K} \right\rfloor + 1. \quad \text{(IV.4)}$$

If $r_{p_1} + T < K$, then

$$\frac{\omega_{p_N}}{K} = \frac{\omega_{p_1}}{K} + \left\lfloor \frac{f_s(T) \cdot x}{K} \right\rfloor; \quad \text{(IV.5)}$$

otherwise

$$\frac{\omega_{p_N}}{K} = \frac{\omega_{p_1}}{K} + \left\lfloor \frac{f_s(T) \cdot x}{K} \right\rfloor + 1. \quad \text{(IV.6)}$$

Algorithm 1 first tries to find the greatest T which satisfies Equation IV.2. Eventually it returns $D_{p_1} = \omega_{p_1} - T - \frac{1}{2}$ as the distance from target point O to probe p_1. Note that if $d_{p_1} \in \mathbb{Z}^+$, we can let the algorithm return $D_{p_1} = \omega_{p_1} - T - 1$ as the detected distance from target point O to probe p_1, where D_{p_1} is the exact value of d_{p_1}.

Thus for correctness, we have the following result which we state formally.

Theorem 4.1: If $D_{p_1} = \omega_{p_1} - T - \frac{1}{2}$ is returned by Algorithm 1, then

$$|D_{p_1} - d_{p_1}| \le \frac{1}{2}.$$

Namely, the error of Algorithm 1 is bounded by $\frac{1}{2}$. In particular, if $d_{p_1} \in \mathbb{Z}^+$, we can find the exact target point with a slight modification to the algorithm by letting the output be $D_{p_1} = \omega_{p_1} - T - 1$.

V. Two-Dimensional Adversarial Methodology

In this section, we prove if the target user is located in the aforementioned probe array, the algorithm will position the target user within an area $\mathcal{S}$ of size $\mathcal{S} \le \frac{\sqrt{3}}{2}$ square meter.

A. Search for the Shadow Point

Before presenting our Two-Dimensional (2-D) algorithm, we summarize the notations introduced throughout the rest of paper in Table III.

TABLE III: Summary of Notations

Symbol	**Meaning**
O'	the target point (the location of the user)
O	the projection of the target point to the line of virtual probes
$d_{p_i} = \|OP_i\|$	the distance from O to P_i
$d_{q_i} = \|OQ_i\|$	the distance from O to Q_i
$D_{p_i} = \|O'P_i\|$	the distance from O' to P_i
$D_{q_i} = \|O'Q_i\|$	the distance from O' to Q_i
ω_{p_i}	the reported distance from O to P_i
ω_{q_i}	the reported distance from O to Q_i

As illustrated in Figure 3, to detect the location of a target user located at position O' on a 2-D plane, we deploy virtual probes along a line close to the target point O'. Let O be the projection of O' to the virtual probe line, we name the probes to the left of O as $P_1, P_2, \cdots, P_N$, and the probes to the right of O as $Q_1, Q_2, \cdots, Q_N$.

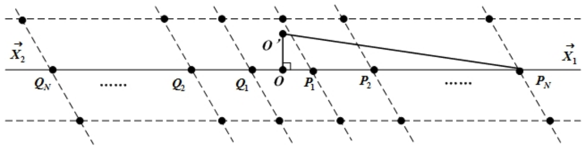

Fig. 3: Projection of The Target Point

We denote the direction from O to P_N as $\vec{X_1}$, and the direction from O to Q_N as $\vec{X_2}$. Similar to the 1-D case, we assume the distance between two adjacent probes is a constant x. We define a problem termed shadow problem detection which is used to find point O, the projection of target point O'. We further assume $|OO'| \le h$ (where $h \le \frac{\sqrt{3}}{2}x$), since O' is close to the line.

We detect the shadow point O from two directions $\vec{X_1}$ and $\vec{X_2}$ respectively. For simplicity, we only apply the direction $\vec{X_1}$ to the following lemmas, which also hold for the direction $\vec{X_2}$.

Lemma 5.1: If $|O'O|$ is bounded, i.e., $|O'O| = h \le \frac{\sqrt{3}}{2}x$, where $x \in \mathbb{Z}^+$ and $\gcd(x, K) = 1$, then

$$|\mathcal{D}_{p_n} - d_{p_n}| \longrightarrow 0, \text{ as } n \longrightarrow \infty.$$

Corollary 5.2: If $n \ge x + 1$, then

$$d_{p_n} < D_{p_n} < d_{p_n} + \frac{1}{2}. \tag{V.1}$$

Lemma 5.3: If $n \ge x+1$, and $0 \le d_{p_1} - \lfloor d_{p_1} \rfloor \le \frac{1}{2}$, then the following holds:

$$\left\lfloor \frac{D_{p_n}}{K} \right\rfloor = \left\lfloor \frac{d_{p_n}}{K} \right\rfloor. \tag{V.2}$$

For any $1 \le T \le K - 1$, let $N = f_s(T) + 1$. We have $d_{p_N} = d_{p_{(f_s(T)+1)}}$, $D_{p_N} = D_{p_{(f_s(T)+1)}}$. Then, we have the following lemma.

Lemma 5.4: There exists T_0, where $1 \le T_0 \le K - 1$, such that for any $1 \le T \le K - 1$, and $T \ne T_0$, we have

$$\left\lfloor \frac{d_{p_{(f_s(T)+1)}}}{K} \right\rfloor = \left\lfloor \frac{D_{p_{(f_s(T)+1)}}}{K} \right\rfloor.$$

Remark 5.5: In fact, Lemma 5.4 is a key component to prove the below Theorem 5.6.

The explicit algorithm to find the shadow point O from the direction $\vec{X_1}$, which is denoted as Algorithm 2 , is demonstrated as follows. Algorithm 2 takes in x, $\{\omega_{p_i}\}$ as input, and returns T and $D_{p_1} = \omega_{p_1} - T - \frac{1}{2}$.

Algorithm 2 Search the Shadow Point from Direction $\vec{X_1}$

```
Initialization
INPUT
The distance between one probe and any adjacent probe: $x$;
The set of samples for the reported distance from Direction $\vec{X_1}$: $\{\omega_{p_i}\}$;
Extended Euclidean Algorithm
    Compute $s$ such that $s \cdot x + t \cdot K = 1$
$T \leftarrow K$
REPEAT
$T \leftarrow T - 1$
    $N = \left(\lfloor \frac{x}{K} \rfloor + 1\right) \times K + f_s(T) + 1$

$\Delta(T) = \begin{cases} 1, & \frac{\omega_{p_N}}{K} = \frac{\omega_{p_1}}{K} + \left\lfloor \frac{f_s(T) \cdot x}{K} \right\rfloor + x \cdot \left(\lfloor \frac{x}{K} \rfloor + 1\right); \\ 0, & \text{Otherwise.} \end{cases}$
UNTIL  $\Delta(T) = 1$  or  $T = 1$
RETURN  $T$  and  $D_{p_1} = \omega_{p_1} - T - \frac{1}{2}$
```

For its correctness, we note that the key role of Algorithm 2 is to test the following equation

$$\frac{\omega_{p_N}}{K} = \frac{\omega_{p_1}}{K} + \left\lfloor \frac{f_s(T) \cdot x}{K} \right\rfloor + x \cdot \left(\left\lfloor \frac{x}{K} \right\rfloor + 1 \right). \tag{V.3}$$

We let

$$M = \left(\left\lfloor \frac{x}{K} \right\rfloor + 1 \right) \times K, \quad \text{and} \quad N = M + f_s(T) + 1,$$

and define

$$\Delta(T) = \begin{cases} 1, & \frac{\omega_{p_N}}{K} = \frac{\omega_{p_1}}{K} + \left\lfloor \frac{f_s(T) \cdot x}{K} \right\rfloor + x \cdot \left(\lfloor \frac{x}{K} \rfloor + 1\right); \\ 0, & \text{Otherwise.} \end{cases}$$

Then we have

$$\frac{\omega_{p_N}}{K} = \frac{\omega_{p_1}}{K} + \left\lfloor \frac{r_{p_1} + T}{K} \right\rfloor + \left\lfloor \frac{f_s(T) \cdot x}{K} \right\rfloor + x \cdot \left(\left\lfloor \frac{x}{K} \right\rfloor + 1 \right). \tag{V.4}$$

If $r_{p_1} + T < K$, then

$$\frac{\omega_{p_N}}{K} = \frac{\omega_{p_1}}{K} + \left\lfloor \frac{f_s(T) \cdot x}{K} \right\rfloor + x \cdot \left(\left\lfloor \frac{x}{K} \right\rfloor + 1 \right); \tag{V.5}$$

otherwise

$$\frac{\omega_{p_N}}{K} = \frac{\omega_{p_1}}{K} + \left\lfloor \frac{f_s(T) \cdot x}{K} \right\rfloor + x \cdot \left(\left\lfloor \frac{x}{K} \right\rfloor + 1 \right) + 1. \tag{V.6}$$

Algorithm 2 first tries to find the greatest T which satisfies Equation V.3. Eventually Algorithm 2 returns output $D_{p_1} = \omega_{p_1} - T - \frac{1}{2}$ as the distance from target point $O^{'}$ to probe p_1. Symmetrically, the explicit algorithm to find the shadow point O from the direction $\vec{X_2}$ is denoted as Algorithm 3.

Above all, the shadow point O can always be located in the 1-meter band by executing the Algorithm 2 and Algorithm 3 respectively.

B. Detect the Shadow Point from Two Directions

In this subsection, we further combine Algorithm 2 and Algorithm 3 that we study before, itemize our deduction and eventually justify if $T + \widetilde{T} = K - 2$, the shadow point O can be uniquely detected from two directions, within 1-meter band accuracy; otherwise $T + \widetilde{T} = K - 1$, the shadow point O can be selected by the middle point returned by either Algorithm 2 or Algorithm 3, of which the accuracy is well below or equal to 1 meter. Hence, the shadow point O can always be pinpointed with an accuracy bounded by 1 meter.

The explicit algorithm to find the shadow point O from direction $\vec{X_1}$ and direction $\vec{X_2}$, which is denoted as Algorithm 4, can be graphically interpreted in Figure 4.

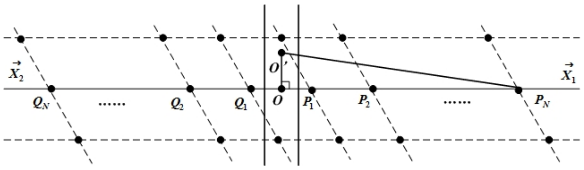

Fig. 4: Detect the Shadow Point from Two Directions

Algorithm 4 takes in x, $\{\omega_{p_i}\}$, and $\{\omega_{q_i}\}$ as input by calling both Algorithm 2 and Algorithm 3 concurrently. By applying Lemma 5.4, Algorithm 4 falls into two decision statements, which are $T + \widetilde{T} = K - 2$, and $T + \widetilde{T} = K - 1$. It eventually returns the setting of the shadow point O on the line segment between Q_N and P_N, which is demonstrated above.

Thus for correctness, we have the following result.

Theorem 5.6: If $D_{p_1} = \omega_{p_1} - T - \frac{1}{2}$, $D_{q_1} = \omega_{q_1} - \widetilde{T} - \frac{1}{2}$ is returned by Algorithm 4, then

$$|D_{p_1} - d_{p_1}| \leq \frac{1}{2}, \quad |D_{q_1} - d_{q_1}| \leq \frac{3}{2}.$$

or

$$|D_{q_1} - d_{q_1}| \leq \frac{1}{2}, \quad |D_{p_1} - d_{p_1}| \leq \frac{3}{2}.$$

Namely, the error of Algorithm 4 is bounded by 1 *meter when outputting the middle point. In particular, if $d_{p_1} \in \mathbb{Z}^+$ or $d_{q_1} \in \mathbb{Z}^+$, we can find the exact shadow point O uniquely with a slight modification to the Algorithm 4 by letting the output be $D_{p_1} = \omega_{p_1} - T - 1$ or $D_{q_1} = \omega_{q_1} - \widetilde{T} - 1$.*

C. Locate Target Point

We first execute Algorithm 4 to detect the shadow point O on the line segment between Q_N and P_N from both direction $\vec{X_1}$ and $\vec{X_2}$. As the probe layout is a lattice of equidistant points, when we make a 60 degrees clockwise rotation for both $\vec{X_1}$ and $\vec{X_2}$, we obtain new directions $\vec{Y_1}$ and $\vec{Y_2}$; when we make a 120 degrees clockwise rotation about direction $\vec{X}$, we obtain new directions $\vec{Z_1}$ and $\vec{Z_2}$. By executing Algorithm 4, we can detect the projected shadow point O_2 along direction $\vec{Y_1}, \vec{Y_2}$, and the projected shadow point O_3 along direction $\vec{Z_1}, \vec{Z_2}$. The rotated projections are graphically illustrated in Figure 5.

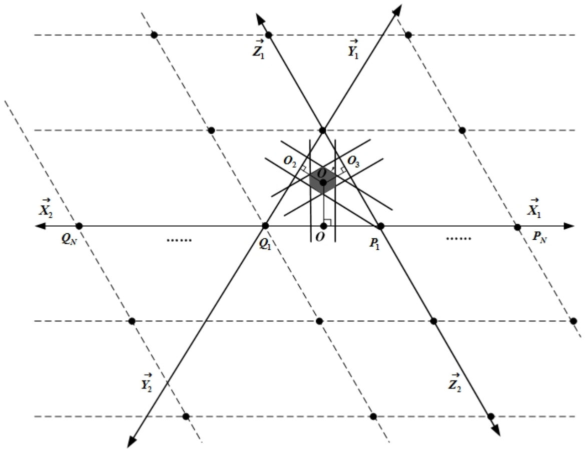

Fig. 5: Overlapping Area

The main algorithm is constructed by concurrently calling Algorithm 4 three times, which is denoted Algorithm 5. Algorithm 5 takes in x, $\{\omega_{p_i}\}$, and $\{\omega_{q_i}\}$ as input, and returns the overlapping area containing the target point $O^{'}$.

Therefore, we can completely detect three shadow points O, O_2, and O_3 respectively from three directions $\vec{X}$, $\vec{Y}$, and $\vec{Z}$ by executing Algorithm 5. Eventually, the target point $O^{'}$ will be located in a zone bounded by $\frac{\sqrt{3}}{2}m^2$. Thus for correctness, we have the following result.

Theorem 5.7: If the target point $O^{'}$ is returned by Algorithm 5, then the overlapping area $\mathcal{S}$ satisfies

$$\mathcal{S} \leq \frac{\sqrt{3}}{2}.$$

Namely, the error of Algorithm 5 is bounded by $\frac{\sqrt{3}}{2}m^2$.

VI. Implementation Discussion

We show the architecture for our attacking methodology in Figure 6. Adversaries gather relative distance samples from the target user's smartphone, when the target user uses LBSD applications. The data are processed in real time via our attacking algorithms and stealthily pinpoint the target user. In Figure 6, the initial s satisfying $s \cdot x + t \cdot K = 1$ is computed via Extended Euclidean Algorithm, which derives Algorithm 1. Algorithm 2 and Algorithm 3 are constructed from two opposite directions $\vec{X_1}$ and $\vec{X_2}$ respectively based on Algorithm 1. Algorithm 4 combines both Algorithm 2 and Algorithm 3 and executes three times from direction $\vec{X}$, $\vec{Y}$, and $\vec{Z}$ respectively, which derives Algorithm 5. Finally, the

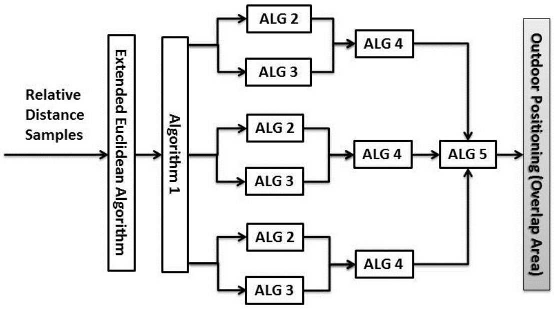

Fig. 6: Implementation of Attacking Model

Algorithm 5 returns the overlapping area containing the target user for tracking or stalking. For each algorithm, if we do Simple Iterative Search, the computation complexity is $\mathcal{O}(K)$, if we do Binary Search, the complexity is $\mathcal{O}(\lfloor\log_2 K\rfloor)$.

In the aforementioned analysis, we assumed that the probe layout satisfies $\gcd(x, K) = 1$. If we generalize the hypothesis such that $\gcd(x, K) = \ell$, where $\ell \in \mathbb{Z}$ and $\ell \geq 1$, the corresponding result will be shown as follows:

- Given $6 \cdot \lfloor\log_2 K\rfloor$ virtual probes, the target user can be pinpointed in an area bounded by $(\frac{\sqrt{3}}{2} \cdot \ell^2)m^2$.

VII. Related Work

The field of location privacy in LBSNs has been scrutinized in recent years. Novel differential privacy mechanisms have also been proposed [5], [6]. There has been quite some research invested in inferring a user's trajectories and further identify his private background information [7], [8]. Zhou *et al.* [9] proposes that an application without any permission may still obtain sensitive information, including a user's geo-location and driving trajectories. The large amount of public background information can potentially turn harmless resources into serious privacy leaks by stealthily monitoring application data-usage statistics, ARP information, etc [10], [11].

In all the aforementioned mechanisms, very little work exists on LBSDs. To the best of our knowledge, the following three approaches are the closest to ours. Le Blond *et al.* [12] discusses that a third party is used to track plenty of users' whereabouts. Both Ding *et al.* [13] and Li *et al.* [3] develop novel automatic tracking systems and demonstrate its effectiveness and efficiency in achieving high-accuracy geo-locating. However, we proceed to quantify its accuracy and develop number theory based algorithms to accurately locate a target user within a small region. Our approach is the first work to prove that a victim can be pinpointed in band-based LBSD services.

VIII. Conclusion and Future Work

To thwart the trilateration attack, contemporary LBSD services have begun to report distances of nearby users in concentric bands. In this paper, we investigated the user location privacy leakage problem in LBSD services reporting distances in concentric bands. Using number theory, we analytically show that by strategically placing multiple virtual probes as pre-determined fake GPS locations, one can nevertheless pinpoint user locations in band-based LBSD. Our methodology guarantees to pinpoint any reported user within an area bounded by one square meter, even for LBSD services using large bands. We emphasize, however, that a person can only be discovered if he is a user of the LBSD service. For example, if a WeChat user only uses WeChat for messaging friends and posting photos, and never uses WeChat's LBSD service, then the user is not locatable by the methods described in this paper. To the best of our knowledge, this is the first work that explicitly exploits and quantifies user location privacy leakage in band-based LBSD services. Our study is expected to draw more public attention to this serious privacy issue and hopefully motivate better privacy-preserving LBSD designs.

Providing other frameworks for purely pinpointing the target user with discrete bands would also be interesting topics. For example, if an attacker has partial information about a set of target users, is it possible to learn the location of all the users? What is the best query to make to pinpoint the target user? We believe these aspects are worth further investigation and leave it as an open direction for future work.

References

[1] "how i was able to track the location of any tinder user". from INCLUDE SECURITY: http://blog.includesecurity.com/2014/02/how-i-was-able-to-track-location-of-any.html.

[2] T. Chen, M. A. Kaafar, and R. Boreli, The where and when of finding new friends: Analysis of a location-based social discovery network. *ICWSM* **(2013)**.

[3] M. Li, H. Zhu, Z. Gao, S. Chen, K. Ren, L. Yu, and S. Hu, All your location are belong to us: Breaking mobile social networks for automated user location tracking. *The 15th ACM International Symposium on Mobile Ad Hoc Networking and Computing* **(2014)**.

[4] V. Shoup, *A computational introduction to number theory and algebra*, Cambridge University Press **(2009)**.

[5] M. E. Andrés, N. E. Bordenabe, K. Chatzikokolakis, and C. Palamidessi, Geo-indistinguishability: Differential privacy for location-based systems. *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*, ACM **(2013)**, pp. 901–914.

[6] P. Shankar, V. Ganapathy, and L. Iftode, Privately querying location-based services with sybilquery. *Proceedings of the 11th international conference on Ubiquitous computing*, ACM **(2009)**, pp. 31–40.

[7] M. Srivatsa and M. Hicks, Deanonymizing mobility traces: Using social network as a side-channel. *Proceedings of the 2012 ACM conference on Computer and communications security*, ACM **(2012)**, pp. 628–637.

[8] H. Zang and J. Bolot, Anonymization of location data does not work: A large-scale measurement study. *Proceedings of the 17th annual international conference on Mobile computing and networking*, ACM **(2011)**, pp. 145–156.

[9] X. Zhou, S. Demetriou, D. He, M. Naveed, X. Pan, X. Wang, C. A. Gunter, and K. Nahrstedt, Identity, location, disease and more: inferring your secrets from android public resources. *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*, ACM **(2013)**, pp. 1017–1028.

[10] A. P. Felt, E. Chin, S. Hanna, D. Song, and D. Wagner, Android permissions demystified. *Proceedings of the 18th ACM conference on Computer and communications security*, ACM **(2011)**, pp. 627–638.

[11] D. Reynaud, E. C. R. Shin, T. R. Magrino, E. X. Wu, and D. Song, Freemarket: Shopping for free in android applications. *Proceedings of the 19th Annual Network & Distributed System Security Symposium* **(2012)**.

[12] S. Le Blond, C. Zhang, A. Legout, K. Ross, and W. Dabbous, I know where you are and what you are sharing: exploiting p2p communications to invade users' privacy. *Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference*, ACM **(2011)**, pp. 45–60.

[13] Y. Ding, S. T. Peddinti, and K. W. Ross, Stalking beijing from timbuktu: A generic measurement approach for exploiting location-based social discovery. *Proceedings of the 4th ACM Workshop on Security and Privacy in Smartphones & Mobile Devices*, ACM **(2014)**, pp. 75–80.