Private Proximity Detection Using Partial GPS Information

Liang Heng, Athindran Ramesh Kumar, and Grace Xingxin Gao, Senior Member, IEEE

Abstract—With the proliferation of GPS-enabled mobile devices, location based services have gained in popularity. In particular, proximity-based services are commonly used to help a user interact with other users nearby. However, these services endanger the privacy of users as the location information is provided to the service provider. This paper proposes a novel private proximity detection scheme based on partial GPS range measurement information. We develop an efficient algorithm for proximity detection and theoretically analyze the false alarm performance. Simulation and empirical results validate this scheme and evaluate its performance.

Index Terms—Location privacy, private proximity detection, location hash, GPS, ranging, location-based services, social network

I. INTRODUCTION

The proliferation of mobile devices with location capability, mainly GPS-enabled smartphones, is revolutionizing modern businesses and lifestyles [1], [2]. Ubiquitous location-aware mobile devices and ever-growing location-based services (LBS) hold considerable promise for enhancing convenience, efficiency, safety, and utility [3]–[5]. Among all LBS, proximity-based mobile social networking, also commonly known as geosocial networking or friend-locator service, is becoming increasingly popular [6]–[8]. A friend-locator service helps a user detect nearby friends. A geosocial networking service facilitates friend-making, i.e., it helps a user find people with matched profile within physical proximity. Examples of these services include Apple’s “Find My Friends”, Facebook’s “Nearby Friends”, Foursquare’s “Radar”, Tencent’s “WeChat”, Ipoki and magnetU. In Find My Friends, a user can see the locations of his friends and get notified when his friends are nearby. In WeChat, a user can find nearby users by the following two ways.

• Shake: A user shakes the phone, and the app will find other WeChat users who are also shaking at the moment locally and around the world. Then the user has an opportunity to message them and make new friends.
• Look Around: Look Around is like Shake without the shaking—the app simply finds other WeChat users who have been recently in the user’s vicinity.

Clearly, all above mentioned LBS require users to share location information to a server (e.g., WeChat) and other users (e.g., Find My Friends). The contextual information attached to user locations reveal the users’ habits, interests, activities, health status, and political and religious affiliations [9]. The high level of intrusion and privacy threats associated has created a reluctance among users to use location-based services [10]–[13]. It would therefore be of great benefit to design practical and effective privacy-preserving proximity detection schemes that reassures users with privacy concerns.

Previous work on location privacy has proposed a variety of methods, among which anonymization, obfuscation, and adding noise are three commonly used techniques [14]–[19]. An underlying concept of these techniques is to degrade location information in a controlled way before releasing it [20]. Comprehensive reviews about the past work in the field of location privacy can be found in [21]–[23].

This paper proposes a new level of location privacy protection, which makes use of the location information inside a GPS receiver. The key difference from previous work (Section II) is that rather than obtaining accurate location information and then degrading it, we extract privacy-preserving location information directly from an intermediate step in GPS location estimation.

In this paper, we present a private proximity detection scheme designed for location-based friend finding in a global social network. Users share their location hash (Section III), which is derived from the user’s GPS range measurements (Sections IV and V), with the server. The server can efficiently detect if any two users are within a threshold distance of each other, but it is computationally intensive for the server to infer each user’s exact location from the location hash. Our scheme can be used independently, or together with other schemes, such as obfuscation, to provide a higher level of privacy protection.

Furthermore, we develop an efficient matching algorithm for proximity detection using the location hash (Section V). We analyze the detection performance and derived an upper bound on probability of false alarm (Section VI). We further conduct extensive evaluations on the performances of our scheme using numerical simulations (Section VII) and experiments (Section VIII). Our evaluation results demonstrated the efficiency and robustness of our scheme for performing private proximity detection.

II. RELATED WORK

The first proximity awareness device (to the best of our knowledge), called Interpersonal Awareness Device (IPAD), was introduced in 1999 [24]. It uses radio signal to measure distance between two devices, and gives aural and visual indications when two devices are close enough (e.g., within 100 meters). As a hand-held device without communication...
capability, it serves as a complement to other forms of communication, such as phone and email. Although simple, IPADs well preserve users’ location privacy.

Atallah et al. [25] studied secure multi-party computational geometry. Their work devised techniques that allow multiple parties (say, Alice and Bob), each having private data (a and b, respectively) to compute some function \( f(a, b) \) without revealing to each other anything unintended. Their protocol lets both Alice and Bob learn whether Bob’s point is in Alice’s polygon. Zhong et al. [26] discuss about potential pitfalls of this protocol, which could compromise user privacy. They proposed three protocols on privacy proximity detection, which achieved the same result as Atallah’s protocol but required less computation and communication. Unfortunately, these methods only work in a distributed social network. Additionally, the total communication cost increases as the square of number of users. Therefore, the computational geometry methods are practical only when the user volume is small.

When there are a large number of users, a centralized server that detects proximity between each pair of users greatly reduces the communication costs. In this scenario, the goal would be to enable the server to perform proximity detection without leaking user locations to the server. The server should be able to infer as less information about user location as possible from the data it is provided with. A possible approach is privacy-preserving test [27], which is based on the location tag initially studied by Qiu et al. in [28], [29]. With proper location tags, location proximity can be reduced to measuring similarity between two sets of tags. Narayanan et al. [27] suggested deriving tags from surrounding environment including WiFi traffic and Access Point identifiers, GSM signals, and GPS signals. Furthermore, Lin et al. used the shingling technique [30], [31] to improve the efficiency of Narayanan’s scheme. The location hash proposed in this paper is closely related to the location tag. A major difference is that the location tag still requires certain types of cryptography to work. The ElGamal encryption suggested in [27] requires much more computational resources on the user end and server end than does our location hash.

This paper is based on and extended from our previous work [32], [33].

### III. Location Hash

#### A. Definition and requirements

Suppose there are \( N \) users at locations \( x_1, \ldots, x_N \). We seek a hash function \( h(\cdot) \) by which the \( n \)th user releases a location hash \( h(x_n) \), \( n = 1, \ldots, N \). The location hash provides the following two levels of privacy protection of the location information \( x_n \).

- **Level 1**: Similar to the cryptographic hash function, a location hash is a one-way function such that it is computationally intensive, if not computationally intractable, to find a solution \( x \) to the equation \( h(x) = h(x_n) \).

- **Level 2**: There are \( M \) solutions, \( M \geq 2 \), to the equation \( h(x) = h(x_n) \), so the probability of correctly guessing \( x_n \) from \( h(x_n) \) is \( 1/M \).

The Level 2 privacy protection will lead to false alarms in proximity detection. Thus, it can be seen as trading some proximity detection accuracy for better privacy protection. Furthermore, the location hash ought to be ephemeral [27]. For example, a location hash expires \( \tau \) seconds after generated. Otherwise, an attacker can precalculate and maintain a lookup table of location hashes so that attacker can invert \( h(\cdot) \) by searching the table.

In addition to the privacy protection capability, the location hash should enable efficient proximity detection. The proximity detection for a threshold distance \( t \) is a Boolean function:

\[
p_t(h(x_n), h(x_m)) = \begin{cases} 
1 & \text{if } |x_n - x_m| \leq t, \\
0 & \text{otherwise}.
\end{cases}
\]

For a global social network with a large number of users, it is necessary that the Boolean function can be efficiently computed.

#### B. Protocol for private proximity detection using location hash

Consider a central server adversary model, in which an untrustworthy LBS server attempts to infer users’ exact location. The location hash defined above will enable private proximity detection against such an adversary.

Suppose in a social network when a user shake his phone, the central server helps him find all other users in his vicinity who shake their phones at the same moment. An example protocol of such a location-privacy-preserving friending service is as follows.

**User end**: User \( n \), \( n \in \{1, \ldots, N\} \) sends his location hash \( h(x_n) \) and interest radius \( t \) to the server whenever he shakes his phone.

**Server end**: The server timestamps every incoming location hash and keeps a cache of recent (within \( \tau \) seconds) location hashes. Suppose the server has a cache of \( K \) recent location hashes, \( \{h(x_{mk})\}_{k=1}^{K} \), upon receiving the friending request with location hash \( h(x_n) \). The server computes the proximity detection function (1) between \( h(x_n) \) and \( h(x_{mk}) \) for all \( k = 1, \ldots, K \). The server replies to user \( n \) with the following set of users who are within the threshold distance \( t \):

\[
\{mk | 1 \leq k \leq K, \quad p_t(h(x_n), h(x_{mk})) = 1\}.
\]

The next two sections introduce GPS fundamentals and describe a practical location hash based on untagged GPS range measurements.

### IV. GPS Basics

The NAVSTAR Global Positioning System (GPS) is a space-based radio navigation system that provides location and time information anywhere on or near the Earth where there is an unobstructed line of sight to four or more GPS satellites [34]. The GPS space segment consists of at least 24 GPS satellites (at the time of writing, 32 [35]) in six orbital planes. Each satellite continuously transmits direct-sequence spread spectrum ranging signals, used to measure the distance between a receiver and the satellite. Ranging signals carry
navigation messages, including ephemeris data and clock data. An ephemeris is used to calculate the position of the satellite, and a clock dataset is used to calculate the satellite clock drift [36]. The GPS receiver uses multilateration to determine its position. As illustrated in Fig. 1, a GPS receiver determines the ranges to at least four satellites by multiplying the speed of light by the time the signal has taken from the satellite to the receiver. The time-of-arrival range measurements are referred to as pseudoranges because they include the receiver clock bias. After compensating for the satellite clock bias and the delay due to troposphere and ionosphere, the pseudorange to the kth satellite can be written as

\[ \rho(k) = \| x(k) - x \|_2 - b + \epsilon(k), \]  

where \( \| u \|_2 = \sqrt{u^T u} \) is the Euclidean norm, the satellite positions \( x(k) \in \mathbb{R}^3 \) are calculated from ephemeris data of the kth satellite, and \( \epsilon(k) \) represents the measurement noise. When pseudoranges to at least four satellites are available, the user receiver position \( x \in \mathbb{R}^3 \) and clock bias \( b \) can be uniquely determined by minimizing

\[ \sum_{k=1}^{K} (\rho(k) - \| x(k) - x \|_2 + b)^2. \]  

To enable code division multiple access, each GPS satellite uses a unique pseudorandom noise (PRN) code to modulate its signal. The satellites are identified by the receiver by means of PRN numbers, so are range measurements.

V. UNTAGGED RANGE MEASUREMENTS

As mentioned in the Introduction, we extract privacy-preserving location information from an intermediate step in GPS location estimation. Our location hash is based on the GPS range measurements described in Section IV. In this scheme, user \( i \) shares his untagged range measurements (without PRN designated)

\[ r_i = [r_i^{(1)}, r_i^{(2)}, \ldots, r_i^{(K_i)}]^T, \]  

where \( K_i \) is the number of satellites visible to this user. The range measurements \( r_i^{(k)} = \rho_i^{(k)} + b, k = 1, \ldots, K_i \), do not include the receiver clock bias \( b \), which can be easily calculated using (3) and removed beforehand. Furthermore, we require \( r_i \) to be a sorted vector in ascending order, i.e.,

\[ r_i^{(1)} \leq r_i^{(2)} \leq \cdots \leq r_i^{(K_i)}. \]

A. Location privacy protection

The untagged range measurements provide Level 1 privacy protection discussed in Section III. When the range measurements are not designated with PRN numbers, if the adversary has no knowledge of satellite orbits, the \( K \) range measurements can be seen as an ordered selection from the \( L \) satellites in the whole constellation. Therefore, the search space is \( K \)-permutations of \( L \). Supposing a user reports \( K = 10 \) range measurements out of a constellation of \( L = 30 \) satellites, the size of search space is \( L!/(L - K)! = 1.1 \times 10^{14} \).

Even though an adversary may use the knowledge of satellite orbits to reduce the search space, there are still a large number of permutations to search. Table I shows the results based on the real GPS orbits (30 active satellites, the same broadcast almanac as used in Section VII-A). When the mask angle is below 5 degrees, a receiver can frequently see more than 10 satellites, which leads to greater search space, although with the orbits information. Because it is computationally intensive for an adversary to infer a user’s actual location from untagged range measurements, the untagged range measurements can protect users’ location privacy, especially in a large social network.

The untagged range measurements can also provide Level 2 privacy protection. First, untagged range measurements seen at two or more distant locations may happen to be similar. These events are categorized as distant false alarms in Section VI. Second, a user can add dummy values to his sorted vector. For example, when the actual range measurements \( [r_i^{(1)}, r_i^{(2)}, \ldots, r_i^{(K_i)}] \), the user can send \( [r_i^{(1)}, \ldots, r_i^{(K_i)}, d_i^{(1)}, d_i^{(2)}]^T \) as his range measurements, where the user drops one actual measurement \( r_i^{(1)} \) and adds two dummies \( d_i^{(1)} \) and \( d_i^{(2)} \). The arbitrary dropped measurements and randomly added dummy values will confuse an untrustworthy server if it tries to find the exact location of user. The dummy values will lead to a lower match ratio in proximity detection described in Section V-C. In this paper, we assume no dummy values in sorted vectors.

Furthermore, the untagged range measurements are ephemeral. The satellite-to-user range is decreasing when the satellite is approaching, and is increasing when the satellite

\[ \begin{array}{c|c}
\text{Mask angle (degrees)} & \text{Search space} \\
\hline
10 & 2.55 \times 10^{13} \\
5 & 2.24 \times 10^{14} \\
0 & 5.13 \times 10^{15} \\
\end{array} \]
C. Blind matching as an optimization problem

Let \subset denote the subset relation between two sorted vectors. We write \( x \subset y \) if each element in \( x \) also belongs to \( y \).

Let \( c \) denote the cardinality, i.e., number of elements, of a vector (or a set) \( x \). The proximity detection problem can be formulated as the following optimization problem:

\[
\begin{align*}
\text{maximize} & \quad c,
\text{subject to} & \quad c = \text{card}(q_1) = \text{card}(q_2), \\
& \quad q_1 \subseteq r_1, \\
& \quad q_2 \subseteq r_2, \\
& \quad \|q_1 - q_2\|_\infty \leq t,
\end{align*}
\]

where the infinity norm \( \|u_1, \ldots, u_n\|^\infty = \max\{|u_1|, \ldots, |u_n|\} \).

The optimization problem maximizes \( c \), the number of matched range measurements. The decision whether the two users are nearby depends on \( c, \text{card}(r_1) \), and \( \text{card}(r_2) \). In this paper, we use a very simple criterion: two users are decided to be nearby if the match ratio

\[
m = \frac{c}{\min\{\text{card}(r_1), \text{card}(r_2)\}} \geq \zeta,
\]

where the decision threshold \( \zeta \), \( 0 \leq \zeta \leq 1 \), is selected to achieve certain detection error performance.

D. Efficient blind matching algorithm

The optimization problem (8) is similar to the longest common subsequence (LCS) problem, which has been extensively studied [39], [40]. Dynamic programming is often used to solve the LCS problem efficiently. Here we borrow this idea to solve the optimization problem.

Let \( r^{(k)} \) denote the \( k \)th element in the sorted vector \( r \), and let \( r[n] \) denote the vector of the first \( n \) elements, i.e., \( r[n] = [r^{(1)}, r^{(2)}, \ldots, r^{(n)}] \). Let \( c(r_1[k_1], r_2[k_2]) \) denote the maximum number of matched range measurements between \( r_1[k_1] \) and \( r_2[k_2] \). We have the following recursive property:

\[
c(r_1[k_1], r_2[k_2]) = \begin{cases} 
0 & \text{if } k_1 = 0 \text{ or } k_2 = 0; \\
\text{c}(r_1[k_1 - 1], r_2[k_2 - 1]) + 1 & \text{if } |r_1^{(k_1)} - r_2^{(k_2)}| \leq t; \\
\text{c}(r_1[k_1 - 1], r_2[k_2]) & \text{if } r_1^{(k_1)} > r_2^{(k_2)} + t; \\
\text{c}(r_1[k_1], r_2[k_2 - 1]) & \text{if } r_1^{(k_1)} < r_2^{(k_2)} - t.
\end{cases}
\]

Using the above property and the fact that both \( r_1 \) and \( r_2 \) are already sorted, we have the following algorithm.

Require: two sorted vectors \( r_i = [r_1^{(i)}, r_2^{(i)}, \ldots, r_n^{(i)}] \), \( i \in \{1, 2\} \)

Require: threshold distance \( t > 0 \)

1: \( k_i \leftarrow 1, i \in \{1, 2\} \)
2: \( c \leftarrow 0 \)
3: while \( k_1 \leq \text{card}(r_1) \) and \( k_2 \leq \text{card}(r_2) \) do
4: if \( |r_1^{(k_1)} - r_2^{(k_2)}| < t \) then
5: \( c \leftarrow c + 1 \)
6: \( k_1 \leftarrow k_1 + 1 \)

The above algorithm achieves the worst-case time complexity of $O(\text{card}(r_1) + \text{card}(r_2))$ and the space complexity of $O(1)$. Since a user usually sees 15 satellites or fewer, $\text{card}(r_1) + \text{card}(r_2)$ is not greater than 30. In practice, the above algorithm is very efficient.

Once we obtain $c$ using the above algorithm, we then use (9) to calculate the match ratio $m$. Two users are determined to be in proximity if $m \geq \zeta$.

VI. PROXIMITY DETECTION PERFORMANCE ANALYSIS

As a statistical hypothesis test, private proximity detection has a probability of making two types of errors: false alarm and missed detection. Suppose there are $N$ users and let $S$ denote the set of all pairs of users, $\text{card}(S) = \binom{N}{2}$. Let $X$ be the set of pairs of users who are within a threshold distance $t$. Let $Y$ be the set of pairs of users who are detected to be close to each other. We define the following performance measures:

- **Probability of false alarm**
  
  $$P_{FA} = \frac{\text{card}(Y \setminus X)}{\text{card}(S \setminus X)},$$  
  \hspace{1cm} (11)

  where the set difference $Y \setminus X = \{ z \in Y \mid z \notin X \}$.

- **Probability of missed detection**
  
  $$P_{MD} = \frac{\text{card}(X \setminus Y)}{\text{card}(X)};$$  
  \hspace{1cm} (12)

- **Probability of detection error**, the combination of missed detections and false alarms
  
  $$P_{DE} = \frac{\text{card}(Y \setminus X) + \text{card}(X \setminus Y)}{\text{card}(S)}.$$  
  \hspace{1cm} (13)

We focus our theoretical performance analysis on $P_{FA}$ for two reasons. First, $P_{DE}$ is usually dominated by $P_{FA}$. In a social network that covers a wide area, most of the user pairs are not in proximity, i.e., $\text{card}(X) \ll \text{card}(S \setminus X) \approx \text{card}(S)$. Therefore, we have

$$P_{DE} = P_{FA} \frac{\text{card}(S \setminus X)}{\text{card}(S)} + P_{MD} \frac{\text{card}(X \setminus Y)}{\text{card}(S)} \approx P_{FA}. \hspace{1cm} (14)$$

Second, the inequality (7) always holds if two users are within the threshold. Therefore, missed detection mainly results from users accidentally losing track of several satellites, which can happen indoors, in an urban canyon, or in other GPS-challenged environments.

It should be noted that there are two types of false alarm:

- **Nearby false alarm (NFA)**: a pair of users are incorrectly detected to be nearby; their actual distance is greater than $t$, but still close to $t$, and they may see the same set of GPS satellites.
- **Distant false alarm (DFA)**: a pair of users are incorrectly detected to be nearby; their actual distance is much greater than $t$, and they may see totally different sets of GPS satellites.

In this paper, nearby false alarm is not our major concern because “proximity” itself is a fuzzy concept in social networking. For example, if two users within a distance $t$ are always deemed nearby, it is acceptable that two users within a larger distance (e.g., 1.5$t$) are detected to be nearby with a certain probability. The following analysis is about probability of distant false alarm ($P_{DFA}$).

A. Probabilistic model of ranges

Suppose we randomly choose a location on the Earth, at a random epoch, the range to an arbitrary GPS satellite observed at this location is a random variable $r$. Let pdf$_r(x) = \frac{d}{\pi} \text{Prob}(r \leq x)$ be its probability density function.

Figure 7 shows the kernel density estimate [41] of ranges from simulation data using real satellite orbits. To simplify our theoretical analysis, we use a uniform distribution to approximate the actual distribution of ranges, i.e., $r \sim U(r_{\text{min}}, r_{\text{max}})$. When the mask angle$^1$ is set to $10^\circ$, $r_{\text{min}} \approx 20189$ kilometers and $r_{\text{max}} \approx 24619$ kilometers. Let the spread of range measurements $\lambda = r_{\text{max}} - r_{\text{min}}$, and we have pdf$_r(x) = 1/\lambda$.

A fundamental assumption of this analysis is that ranges to different satellites are independent and identically distributed (i.i.d.). The validity and efficacy of this assumption has been demonstrated in our previous work [42].

In this analysis, we ignore GPS range measurement errors because such errors are much less than the threshold distance.

B. Probability of false alarm

Suppose user 1 reports an untagged range vector $r_1 = [r_1^{(1)}, \ldots, r_1^{(K_1)}]^T$ and user 2 reports an untagged range vector $r_2 = [r_2^{(1)}, \ldots, r_2^{(K_2)}]^T$. Both users are randomly chosen on the Earth so that with a very high probability, the two users are far apart. Let $c$ denote the number of matched range matched range measurements. According to our discussion in Section V-B, false alarm occurs when the match ratio $m = c / \min\{K_1, K_2\}$ is greater than or equal to the threshold $\zeta$.

Let us randomly shuffle $r_2$ so that its elements are no longer sorted. Then, based on our i.i.d. assumption above, $r_2^{(k)} \sim U(r_{\text{min}}, r_{\text{max}})$ for all $k = 1, \ldots, K_2$. The probability of $r_2^{(1)}$ matching one of the elements of $r_1$ is given by

$$\text{Prob}(r_2^{(1)} \in \bigcup_{k=1}^{K_1} \left[ r_1^{(k)} - t, r_1^{(k)} + t \right]) \leq \sum_{k=1}^{K_1} \text{Prob}(r_2^{(1)} \in \left[ r_1^{(k)} - t, r_1^{(k)} + t \right]) \hspace{1cm} (15)$$

If $r_2^{(1)}$ matching one of the elements of $r_1$, then the probability of $r_2^{(1)}$ matching one of the remainder elements of $r_1$ has a

$^1$Mask angle, also referred to as cutoff angle, is a parameter in GPS antenna design and GPS receiver configuration for excluding low-elevation satellites from position solution.
similar upper bound $2t(K_1 - 1)/\lambda$. Similarly, the $i$th match happens with a probability less than or equal to $2t(K_1 + 1 - i)/\lambda$.

Let $\eta = \lceil \zeta \min\{K_1, K_2\} \rceil$ be the required number of matched range measurements, where $\lceil \cdot \rceil$ is the ceiling function. Finally, the probability of at least $\eta$ matched range measurements has the following upper bound:

$$P_{DFA} = \text{Prob}\{m \geq \zeta \text{ two randomly chosen users}\} = \text{Prob}\{c \geq \eta \text{ two randomly chosen users}\} \leq \frac{2tK_1}{\lambda} \frac{2t(K_1 - 1)}{\lambda} \ldots \frac{2t(K_1 + 1 - \eta)}{\lambda} \frac{K_2}{\eta} \quad (16)$$

$$= \eta! \left(\frac{2t}{\lambda}\right)^{\eta} \left(\frac{K_1}{\eta}\right) \frac{K_2}{\eta} .$$

Since $P_{DFA} \leq 1$, we finally have the following upper bound:

$$P_{DFA} \leq \min\left\{1, \eta! \left(\frac{2t}{\lambda}\right)^{\eta} \left(\frac{K_1}{\eta}\right) \frac{K_2}{\eta}\right\} . \quad (17)$$

The equation clearly shows that increasing $\lambda$ (equivalent to using lower mask angle) and/or decreasing $t$ will reduce false alarm rate.

Figure 3 shows the upper bound on $P_{DFA}$ calculated using (17) for $K_1 = K_2 = 10$, different decision threshold $\zeta$, and different threshold distance $t$. As expected, increasing $\zeta$ will reduce false alarm rate because a higher $\zeta$ means that more matched range measurements are required. In addition, increasing $t$ will increase false alarm rate. In practice, the threshold distance is usually below 10 kilometers, and our method can achieve $\leq 10^{-5}$ false alarm rate if $\zeta \geq 0.5$.

Figure 4 shows the upper bound on $P_{DFA}$ versus decision threshold $\zeta$ for threshold distance $t = 1000$ meters and different number of visible satellites $K_1$ and $K_2$. Again, increasing $\zeta$ will reduce falsealarm rate. For a fixed $\zeta$, more visible satellites yield lower false alarm rate.

Figure 5 is similar to Fig. 4, but the difference is that we fixed $K_1$ to be 10 and change $K_2$ from 6 to 14. Comparing Fig. 4 and Fig. 5, we can see that unequal length of untagged

range vectors deteriorates performance. For example, the false alarm rate when $K_1 = 10$ and $K_2 = 6$ is always higher than that when $K_1 = K_2 = 6$, and the false alarm rate when $K_1 = 10$ and $K_2 = 14$ is always higher than that when $K_1 = K_2 = 10$.

In summary, the upper bound on $P_{DFA}$ is a function of the spread of range measurements $\lambda$, the decision threshold $\zeta$, and the number of satellites seen by each user, $K_1$ and $K_2$. $P_{DFA}$ is lower for a larger $\lambda$ (i.e., a lower mask angle), a lower distance threshold $t$, a higher decision threshold $\zeta$, a greater $\min\{K_1, K_2\}$, a lower $|K_1 - K_2|$, or any combination of these.

VII. SIMULATION RESULTS

This section conducts numerical simulations to validate our theoretical analysis and to evaluate the proximity detection
performance. Our simulations are based on satellite orbits computed from real almanacs broadcast from GPS satellites. We assume perfect signal reception, i.e., no occasional loss of GPS signals from low-elevation satellites.

A. Simulation settings

- Satellite orbits: GPS broadcast almanacs on 6 September 2014.
- Total number of GPS satellites in service: 30.
- Mask angle: $10^\circ$.
- Range measurement errors: i.i.d. zero-mean Gaussian distribution with standard deviation of 5 meters.
- Spatial sampling modes:
  - Global: 1807 nodes evenly distributed on the earth (average distance between two neighboring nodes: 556 kilometers), as shown in Fig. 6;
  - Local: 801 $\times$ 801 nodes in a square centered at Urbana, IL (average distance between two neighboring nodes: 278 meters).
- Temporal sampling (global mode): 1 sample per 46 minutes over 1 day (31 epochs in total).

B. Global mode

As shown by Fig. 6, the global mode simulate 1807 users all over the world. Since all nodes are far apart (the distance between any two nodes is at least 500 kilometers), no pairs should be detected to be in proximity. Figure 7 shows the kernel density estimate [41] of range measurements observed at these nodes, overlapped with the uniform distribution used in our theoretical analysis. The purpose of this simulation is to correct possible errors in the theory and to provide a more realistic assessment.

The simulation computes the maximum number of matched range measurements, $c$, for all $\binom{1807}{2}$. The whole procedure is as follows.

1: for each of the 31 epochs do
2: compute satellite positions using broadcast almanacs;
3: for each of the 1807 nodes do
4: compute noiseless range measurements to visible satellites;
5: add Gaussian noises to range measurements;
6: for every pair of node do
7: compute match ratio;

Figure 8 shows relative frequency of the number of visible satellites. It is most frequent for a user to see 10 satellites. With more than 90% of the time, a user can see 8 satellites or more.

Figure 9 shows the probability of false alarm $P_{FA}$ versus threshold distance $t$ for different decision threshold $\zeta$. The curves shows an approximate linear relationship between $\log(P_{FA})$ and $\log(t)$. The theoretical upper bounds plotted in Fig. 3 show a similar linear relationship. It should be noted that we cannot compare the values across the two figures because Fig. 3 is based on $K_1 = K_2 = 10$, while Fig. 9 includes all $(K_1, K_2)$ pairs in the simulation.
Figure 9. Probability of false alarm $P_{FA}$ versus threshold distance $t$ for different decision threshold $\zeta$. The trend of the curves is consistent with our theoretical results in Fig. 3.

Figure 10. Probability of false alarm $P_{FA}$ versus decision threshold $\zeta$ for different number of visible satellites ($K_1 = K_2$), compared with the theoretical upper bounds. The threshold distance $t$ is fixed to 1000 meters.

Figure 11. Number of matched range measurements for the whole region. Distant false alarm rate is zero if we choose a decision threshold $\zeta \geq 4/7$.

Figure 12. Number of matched range measurements $c$ for the whole central area. It can be seen that the $c$ values form a geometric pattern, which reflects the geometry of the GPS satellites at the epoch. Fig. 11 shows that distant false alarm rate is zero if we choose a decision threshold $\zeta \geq 4/7$. Fig. 12 shows that all points in the circle with a radius of $t$ have at least 6 matched range measurements. Therefore, missed detection rate is zero if we choose a decision threshold $\zeta \leq 6/7$. Fig. 12 also shows that the points in the circle with a radius of $2t$ have at least 4 matched range measurements. Thus, nearby false alarm occurs if the decision threshold $\zeta \leq 4/7$.

VIII. Empirical Results from Real GPS Receiver Outputs

A. Experiment with a global network of static GPS receivers

We further validate our theory and algorithm using real GPS pseudorange measurements collected by the International GNSS Service (IGS) [43] and the University NAVStar Consortium (UNAVCO). The two networks consist of more than 1000 stations all over the world. Each station has one or multiple GPS receivers continuously generating GPS pseudorange measurement data. The pseudoranges still have the user clock bias in them. We removed the clock biases using the method mentioned in Section 4.3.2 of [44].

We treat the IGS and UNAVCO stations around the world as nodes to test for proximity using the scheme outlined in Section V. These stations are usually very far (at least tens of kilometers) apart. With a proper distance threshold, the simulation procedure is the same as that for the global mode, except that in the local mode, we only compute the match ratio between the central node and every other node.

We choose a threshold distance $t = 1000$ meters. We choose a relatively bad epoch that the center node can see only 7 GPS satellites. We expect $c$ to be close to 7 for all nodes within 1000 meters, and to 0 for all other nodes, especially the nodes faraway.

Figure 11 shows the number of matched range measurements $c$ for the whole region, and Fig. 12 zooms in on the central area. It can be seen that the $c$ values form a geometric pattern, which reflects the geometry of the GPS satellites at the epoch. Fig. 11 shows that distant false alarm rate is zero if we choose a decision threshold $\zeta \geq 4/7$. Fig. 12 shows that all points in the circle with a radius of $t$ have at least 6 matched range measurements. Therefore, missed detection rate is zero if we choose a decision threshold $\zeta \leq 6/7$. Fig. 12 also shows that the points in the circle with a radius of $2t$ have at least 4 matched range measurements. Thus, nearby false alarm occurs if the decision threshold $\zeta \leq 4/7$. 

C. Local mode

The local mode simulates $801 \times 801 = 641,601$ nodes in a 222 kilometers-by-222 kilometers square. The distance between two neighboring nodes is 278 meters. The purpose of local mode simulation is to evaluate the probability of nearby false alarm and the probability of missed detection.

We calculate the number of matched range measurements, $c$, between the central node and all other 641,600 nodes. The
receivers at different stations can be seen as distant users, while the receivers at the same stations are nearby users.

We applied our algorithm to the IGS data recorded on 10 January 2014. The pseudorange measurements released by 1171 stations around the world at the start of the day at one time epoch was used to aggregate the statistics for validation. For each pair of stations, we computed the number of matched range measurements \( c \) as defined in (8) and the match ratio \( m \) as defined in (9). Fig. 13 shows the relative frequency of the number of visible satellites from the IGS stations at the chosen time epoch. The number of visible satellites is around 9 on the average, lower than that in Fig. 9. This shows that real GPS receivers may lose satellites.

In Fig. 14, we plot match ratio \( m \) as a function of actual distance for a threshold of \( t = 5000 \) meters. We can clearly see that the match ratio is high for stations which are within the threshold distance \( t \). The match ratio is also high for some pairs whose actual distance is between \( t \) and \( 3t \), causing nearby false alarms.

Figure 15 shows the variation of probability of false alarm with the threshold distance \( t \). Low false alarm rate \( (P_{FA} \leq 10^{-4}) \) is achieved when \( \zeta \geq 0.7 \) for threshold distance \( t \leq 5000 \) meters. When \( t \geq 5000 \) meters, \( P_{FA} \) is dominated by distant false alarms. Thus, we see an approximate linear relationship between \( \log(P_{FA}) \) and \( \log(t) \), similar to the theoretical upper bounds. When \( t \leq 2000 \) meters, distant false alarms occur less frequently. Therefore, nearby false alarms dominate \( P_{FA} \) and lead to an error floor on the order of \( 10^{-6} \).

Figure 16 shows the variation of probability of missed detection with the threshold distance \( t \). It can be seen that the missed detection rate is below 0.05 for decision threshold \( \zeta \leq 0.8 \). However, the false alarm rate is obviously higher for lower values of \( \zeta \). This trade-off is succinctly depicted in Fig. 17, which plots the \( P_D = 1 - P_{MD} \) versus \( P_{FA} \), also
Fig. 16. Probability of missed detection $P_{MD}$ versus threshold distance $t$. Low missed detection rate ($P_{MD} \leq 0.05$) is achieved when $\zeta \leq 0.8$ for all threshold distance.

Fig. 17. Receiver operating characteristic (ROC) curve.

known as the receiver operating characteristic (ROC) curve [45].

The results with real data demonstrates robustness of our scheme. Occasional loss of satellites cause missed detection. With a proper choice of decision threshold $\zeta$, we can still achieve detection performance of $P_{FA} \leq 10^{-4}$ and $P_{MD} \leq 0.05$, which is satisfactory for many social network applications.

B. Experiment with four portable GPS receivers

With the evaluation using the IGS tracking network, the “user” locations were fixed. Further, most pairs of stations were very far apart and the locations of the stations in the tracking network do not model the distribution of mobile phone users very well. Thus, we perform some local experiments to further validate the algorithm.

In our experiment, pseudorange measurements were collected using u-blox LEA-6T receivers. Four students from University of Illinois took part in this experiment. Two of them drove in opposite directions from the Urbana-Champaign campus for a few kilometers. Two others were walking on the campus. The paths of these users are presented in Fig. 18.

Figures 19, 20 and 21 present the variation of match ratio with distance for $t = 750$, 1500, and 5000 meters, respectively, for a pair of users. The distance between the two users are based on their GPS position outputs. The match ratio is based on pseudorange measurements from the receivers. From Fig. 19, we can see that there is a sharp decrease in match ratio when the distance between the users increases more than the threshold of $t = 750$ meters. This demonstrates the robustness of the algorithm. For the same pair of users, as can be seen in Fig. 20 that, a higher threshold of $t = 1500$ meters gives appropriate results in terms of match ratio. In Fig. 21, the threshold $t = 5000$ meters is always above the distance between the two users. As a result, the match ratio is close to 1 most of the time.

In this experiment, the users frequently experienced bad GPS conditions in the urban environment, and their fluctuant position solutions led to spikes in the “Distance” curves.
in Figs. 19 to 21. Our private proximity detection scheme demonstrates reasonably good robustness in such a GPS-challenging environment.

Due to noisy pseudorange measurements, the decision threshold $\zeta$ should be lower for shorter threshold distance. Figs. 19 to 21 suggest that the best decision threshold $\zeta = 0.65$, 0.7, and 0.8 for threshold distance $t = 750$, 1500, and 5000 meters, respectively. For threshold distances lower than 500 meters, a lower decision threshold $\zeta$ can be used to tolerate large range errors. Figs. 3 and 10 have shown that a lower decision threshold can still achieve good $P_{FA}$ at lower threshold distances.

IX. CONCLUSION

This paper proposed a novel private proximity detection method, which makes use of partial GPS measurement information. We developed an efficient algorithm for proximity detection. We theoretically analyzed proximity detection performance and derived an upper bound on probability of false alarm. Our numerical simulations based on real satellite orbits validated our algorithm and demonstrated its potential usefulness. We further conducted experiments using data from a global network of GPS receivers. The empirical results demonstrated the efficiency and robustness of our scheme for performing private proximity detection in mobile social networking.

REFERENCES


