Revisiting Localization Attacks in Mobile App People-Nearby Services

Jialin Wang[1], Hanni Cheng[1], Minhui Xue[2,3], and Xiaojun Hei[1(✉)]

[1] Huazhong University of Science and Technology, Wuhan 430074, China
{wangjialin,chenghn,heixj}@hust.edu.cn
[2] East China Normal University, Shanghai 200062, China
[3] NYU Shanghai, Shanghai 200122, China
minhuixue@nyu.edu

Abstract. The widespread use of people-nearby services has spawned the development of social discovery applications that help users make new friends with nearby users (such as WeChat). Unfortunately, malicious third-parties can often deploy trilateration attacks to exploit people-nearby applications to determine the exact locations of target users, therefore compromising their privacy. In this paper, we revisit these localization attacks and propose a new two-step localization method that boosts the accuracy of the state of the art for the contemporary location-based social network (LBSN) services which have adopted the band-distance obfuscation to blur the location information. The basic idea is to first locate the target in a small circle with the radius of the band distance; then, refine the estimated location with sufficient queries which is driven by the required localization accuracy. We theoretically prove that our method is able to converge to pinpoint users with an upper bound of the complexity of our design. We also evaluate the performance of our model when considering different distribution errors, and finally show our localization method is robust with exciting accuracy and limited complexity through extensive simulation experiments. This attack can locate target users within 20 m with over 95% accuracy in most cases while the query-time is a limited value and can be roughly computed.

Keywords: Privacy leakage · Localization attack
Two-step localization · Location-based social network · WeChat

1 Introduction

Location-based services (LBS) provide value-added applications for users based on their locations. LBS can be used in a variety of contexts, such as health, indoor object search, entertainment, work, personal life, etc. Internet applications such as Facebook and Yelp allow users to "check-in" at restaurants, bars, retail outlets, schools, and offices, thereby sharing their locations within their social networks. However, the appearance of "the Snowden incident" has risen the people's

G. Wang et al. (Eds.): SpaCCS 2017, LNCS 10656, pp. 17–30, 2017.
https://doi.org/10.1007/978-3-319-72389-1_2

consciousness of the privacy protection. Internet users are being reminded frequently that their online behaviors have been under constant scrutiny by NASA and other third parties.

All these factors have contributed to the recent growth of privacy-preserving mobile application people-nearby services. Applications with people-nearby services utilize the users' geographical information to provide proximity-based social and message discovery. People-nearby services are being used to find dating partners, friends who live or work nearby, and for bulletin-board style messages that have been posted nearby. These recent-growing people-nearby services may based on anonymous messaging or relatively close relationship: 4chan, Whisper, and Yik Yak that allow users to anonymously post their thoughts to a public audience; and WeChat that allows users to share content only visible to friends.

The above cases of privacy-preserving communications have brought forth a challenging privacy problem: **Can a malicious third-party determine the locations of those nearby users by any people-nearby services in mobile application?** News articles have reported that the Egyptian government used trilateration to locate and imprison users of gay dating applications [1,2]. Recent academic work has also shown several general avenues of localization attacks to evaluate privacy of the following two types of mobile application nearby services with *exact-distance* and *band-distance*. (i) If a mobile application people-nearby service provides the exact distance to the nearby user or message, the attack can be launched by taking readings from at least three different vantage points to trilaterate or triangulate the user's location [3,4]. The readings from different vantage points can be set by virtual probes using fake GPS locations. (ii) If the mobile application people-nearby services do not provide an exact distance to the nearby user or message but instead report distances of nearby users in concentric bands, such as bands of 100 m as used by WeChat. Other research can still infer user's location with high accuracy from theory to practice [3–10].

In this work, we revisit mobile app people-nearby services with band distances and adopt a new model that is different from the existing methods to localize a user in a two-dimensional plane. In theory, our method can localize the target user within a square of any size. We also derive a complexity upper-bound of our algorithm. In practice, we acknowledge that localization errors may occur because of the GPS measurement deviation or that errors are intentionally added by mobile apps for privacy protection. At the same time we evaluate the performance of our model when considering different distribution errors, and finally show our localization method is robust through extensive simulation experiments.

The paper is organized as follows. Section 2 reviews some related work. Section 3 demonstrates the localization method. In Sect. 5, we evaluate our model based on simulation experiments considering different error distributions. In Sect. 6, we present some discussions. Finally, Sect. 7 concludes the paper.

2 Related Work

There have been quite a few studies on localizing users using mobile app people-nearby services either with exact distances or band distances. In Euclidean geometry, trilateration is the process of determining relative locations of points by measurement of exact distances, using the geometry of circles; triangulation is the process of determining the location of a point by measuring angles to it from known points at either end of a fixed baseline, rather than measuring distances to the point directly (trilateration). To perform the trilateration attack, assume that when a target user is known to lie on three circles from known locations, the centers of the three circles with their exact radii provide sufficient information to pinpoint the location of the target user [11]; the triangulation attack is similar. Qin *et al.* demonstrated triangulation attacks against services with exact distances [12]. In other independent studies, Mascetti *et al.* [13] applied a distance-based clustering algorithm to formalize a location privacy attack to approximately localize the users. Li *et al.* [3] explored user discovery attacks and highlight the significance of this threat. However, the method has some limitations that it may require applications' information while it is sensitive to the noise introduced by nearby services. Polakis *et al.* [4] conducted a theoretical study and proved tight bounds on the number of queries required for carrying out the localization attacks, irrespective of machine learning techniques.

Recently, many measurement studies focus on online social networks such as Whisper [5,14], WeChat [6–10,15,16], and Yik Yak [17]. These online social networks have stored large volumes of sensitive data about users (e.g., controversial discussion information, user profiling, activity traces), all of which pose potential privacy risks.

3 A Two-Step Localization Model

Most locating models proposed in recent literature focus on the practical effects on the application people-nearby services while few of them have the theory analysis for the localization errors and the complexity of their models. In addition, the experiment results show that there are still room for improving the localization accuracy. In this section, we propose a new localization attack model in which localization errors can be quantified. Based on the theoretical analysis, we find an efficient way to locate the target user and quantify the query complexity of our model. Table 1 tabulates the notations that will be used throughout the rest of this paper.

3.1 A Two-Step Localization Algorithm

Since the location information provided is not specific enough to locate the target user easily with high precision, we divide the locating procedure into the following two steps: (i) Coarsely locate the target to a small circle whose radius is r, which needs to traverse the whole target distributing ring. This step is called

Table 1. Notations

Notation	Description
$dist(A, V)$	The Euclidean distance between point A and V
r	The band distance of the application
V	The target point
A_i	The attacker
d_r	The reporting distance of the application

LocateToR; (ii) Restrict the user within a small square precisely and efficiently when constraints to the horizontal and vertical coordinates are added. The step is called *LocateAccurate*. Algorithm 1 describes the entire locating procedure.

Algorithm 1. Two-step localization

Input:
r: the band distance of the mobile app
ϵ: the tolerant error
Output:
p_{est}: estimating target position
% the whole locating procedure
1: $p = LocateToR(r)$
2: $p_{est} = LocateAccurate(r, p, \epsilon)$
3: **return** p_{est}

3.2 Coarse Location Estimation

LocateToR is aimed at coarsely restricting the target to a small circle whose radius is r. The basic idea is to cover the target distribution ring with the circle one by one until the reporting distance is r which indicates that the target is inside the circle, and then record the center of the circle for the following accurate localization.

The whole implementation is described in Algorithm 2. Suppose that the band distance of the application is r and the target is V, the reporting distance from V to initial attacker location A_0 is d_r. To traverse the target circular band, we set all the covering circle centers distributing along the circle whose radius is $d_r - \frac{r}{2}$. We can easily take A_1 that is $d_r - \frac{r}{2}$ far above the A_0 as shown in Fig. 1. The coordinates of A_i can be computed according to the angle $\angle A_1 A_0 A_i$ and the coordinate of A_0. Besides, the θ in the figure can be calculated by the cosine formula (take the advantage of triangle $A_1 A_0 D$ in the illustration). As for *LocateToR*, we prove the following theorem.

Algorithm 2. LocateToR

Input:

r: the band distance of the app
p: the initial attacker position

Output:

p: the coordinates of point which is less than r far from target
% restrict the target to a circle of radius r

1: $d_r = AppDist(p)$
2: $p = AttackerInit(p)$
3: **while** $d_r > r$ **do**
4: $p = NextAttacker(p, r)$
5: $d_r = AppDist(p)$
6: **end while**
7: **return** p

Theorem 1. In a two-dimensional space, given a point V existing in a ring with internal diameter $d_A - r$ and external diameter d_A, if we cover the ring with the disk whose radius is r. There must be a disk and the distance between its center and V is less than r, and it takes at most Ω queries to find the disk, which satisfies $\Omega = \frac{1}{2} \cdot \frac{2\pi}{2\theta} = \frac{\pi}{2arccos\left(\frac{(d^2+(d-\frac{r}{2})^2-r^2)}{2d(d-\frac{r}{2})}\right)}$.

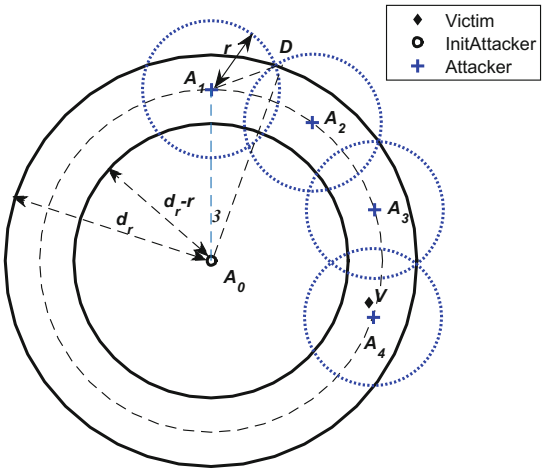


Fig. 1. An illustration of the LocateToR algorithm

3.3 Location Refinement

In *LocateToR*, we have restricted the target point V in a small circle of radius r, now we turn our attention to locating V accurately. In Fig. 2, $P \in \{W, E, N, S\}$

is r away from the target V. We reach our goal by finding these four points P that satisfy $dist(\hat{P}, P) < \epsilon$ on the line which is horizontal or vertical. ϵ is the tolerance set by us. It seems that V is pulled by P from four directions. Thus, the target V is restricted in a small square $C_0C_1C_2C_3$ whose length of side is ϵ. We take the center (i.e., V_{est}) of the square to be the estimated position of the target V. The algorithm details are shown in Algorithm 3.

Now we concentrate on how to determine $\hat{W}, \hat{E}, \hat{N}, \hat{S}$ very fast. We first prove Theorem 2, Corollaries 1 and 2. *BiSecSearch* (i.e., Algorithm 4) is used to search the eligible points. Starting from the position which is returned by *LocateToR*, we hunt for $\hat{W}, \hat{E}$ first. The moving distance is r at the beginning and reduces by half every time when moving forward to the actual point P. Once the reporting distance changes, moving direction will be changed. From Corollary 2, we make sure that $dist(P, \hat{P})$ decreases exponentially. Then, taking the central point of $\hat{W}, \hat{E}$ as the starting point, we obtain $\hat{N}, \hat{S}$ in the similar way of determining $\hat{W}, \hat{E}$.

Algorithm 3. LocateAccurate

Input:
r: the band distance of the mobile app
p: the point returned by LocateToR
ϵ: the tolerant error

Output:
p_{est}: estimating target position
% locate the target accurately

1: $[p_{X1}, p_{X2}] = BiSecSearch(r, p, X, \epsilon)$
2: $p = \frac{p_{X1}+p_{X2}}{2}$
3: $[p_{Y1}, p_{Y2}] = BiSecSearch(r, p, Y, \epsilon)$
4: $p_{est,x} = \frac{p_{X1,x}+p_{X2,x}}{2}$
5: $p_{est,y} = \frac{p_{Y1,y}+p_{Y2,y}}{2}$
6: **return** p_{est}

Theorem 2. For any two points A, V in a two-dimensional space, the distance between them satisfies $dist(A, V) < r$, and the two different points A_1,A_2 can be found on any straight line crossing the A point, which satisfies the following equations:

$$dist(A_1, V) = r,$$
$$dist(A_2, V) = r.$$

Corollary 1. For any two points $A(x_A, y_A)$, $V(x_V, y_V)$ in a two-dimensional space, the distance between two points satisfies $dist(A, V) < r$. Then two different points A_1 and A_2 can be found on the line $x = x_A$ *or* $y = y_A$ that is parallel to the axis through the point A, which satisfies:

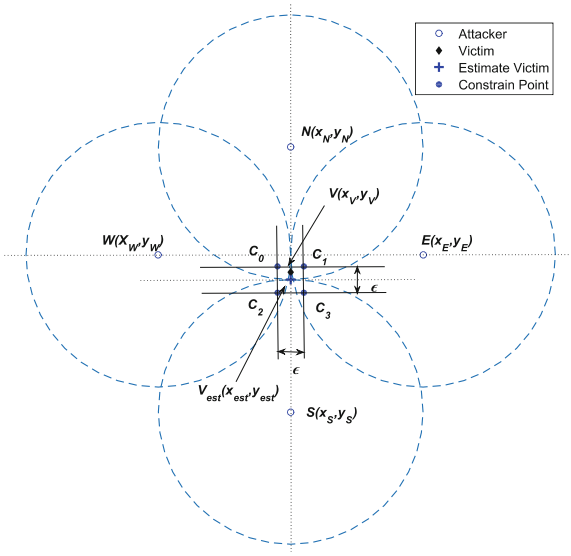


Fig. 2. An illustration of the location refinement algorithm

$$dist(A_1, V) = r,$$
$$dist(A_2, V) = r,$$
$$\frac{x_{A_1} + x_{A_2}}{2} = x_A \ or \ \frac{y_{A_1} + y_{A_2}}{2} = y_A.$$

Corollary 2. For any two points A, V in a two-dimensional space, the distance between two points satisfies $dist(A, V) < r$, there must exist A_1, A_2 satisfies the Corollary *1*. Starting from A, move the point along the any line that crosses the A point with the moving sequence $l = \{l_i | l_i = \frac{r}{2^i}, i = 0, 1, \ldots, N-1\}$ and the direction is uncertain. For any small ϵ, after N times of movement, we can find $\hat{A}_1$ *and* $\hat{A}_2$ respectively, which satisfies*:*

$$dist(A_1, \hat{A}_1) < \epsilon,$$
$$dist(A_2, \hat{A}_2) < \epsilon,$$

where

$$N = \left\lceil \log_2 \frac{\sqrt{2}r}{\epsilon} \right\rceil + 1.$$

4 Theoretical Analysis

In Fig. 2, the target V is at (x_{victim}, y_{victim}), and we denote the estimation point with $Vest(x_{est}, y_{est})$. On the line $y = y_{victim}$, $W(x_W, y_W)$ and $E(x_E, y_E)$

Algorithm 4. BiSecSearch

Input:

r: the band distance of the mobile app
p: the initial point
dim: X or Y
ϵ: the tolerant error when locate the point

Output:

P: List of the points of estimated position along the desired direction
% look for the point who is r away from the target

$IterNum = \lceil log_2 \frac{\sqrt{2}r}{\epsilon} \rceil + 1$
P is a empty list
for sgn=-1,1 **do**
  $p_{e,dim} = p_{dim} - sgn * r$
  $dim_2 = \{X, Y\} \backslash dim$
  $p_{e,dim_2} = p_{dim_2}$
  for $j = 0, 1, ..., IterNum$ **do**
    $d_r = AppDist(p_e)$
    if $r > d_r$ **then**
      $p_{e,dim} = p_{e,dim} + sgn * r$
    else
      $p_{e,dim} = p_{e,dim} - sgn * r$
    end if
  end for
  insert p_e into P
end for
return P

are the points satisfying Corollary 1, $\hat{W}(\hat{x_W}, \hat{y_W})$ and $\hat{E}(\hat{x_E}, \hat{y_E})$ are the points satisfying Corollary 2, ϵ is the tolerance parameter in *LocateAccurate*.

$$\sqrt{(x_W - x_{victim})^2 + (y_W - y_{victim})^2} = r, \tag{1}$$

$$\sqrt{(x_E - x_{victim})^2 + (y_E - y_{victim})^2} = r. \tag{2}$$

According to Corollary 2,

$$\sqrt{(x_W - \hat{x_W})^2 + (y_W - \hat{y_W})^2} < \epsilon.$$

Since $y_W = y_E = y_I = \hat{y_W} = \hat{y_E}$, then

$$|x_W - \hat{x_W}| < \epsilon, \tag{3a}$$

$$|x_E - \hat{x_E}| < \epsilon, \tag{3b}$$

in which

$$N = \left\lceil \log_2 \frac{\sqrt{2}r}{\epsilon} \right\rceil + 1.$$

Suppose ϵ_W, ϵ_E is the true error when estimating x_W, x_E, i.e.,

$$\epsilon_W = |x_W - \hat{x_W}| < \epsilon, \tag{4a}$$

$$\epsilon_E = |x_E - \hat{x_E}| < \epsilon. \tag{4b}$$

According to Corollary 1,

$$\left| \frac{x_W + x_E}{2} - x_{victim} \right| = 0. \tag{5}$$

Then, we have

$$\begin{aligned} |x_{est} - x_{victim}| &= \left| \frac{\hat{x_W} + \hat{x_E}}{2} - x_{victim} \right| \\ &= \left| \frac{(x_W - \epsilon_W) + (x_E - \epsilon_E)}{2} - x_{victim} \right| \\ &= \left| \frac{x_W + x_E}{2} - x_{victim} + \frac{\epsilon_W + \epsilon_E}{2} \right| \\ &= \left| \frac{\epsilon_W + \epsilon_E}{2} \right| < \frac{|\epsilon_W| + |\epsilon_W|}{2}. \end{aligned} \tag{6}$$

If $\epsilon < \frac{r}{2^{N-1}}$, we have

$$|x_{est} - x_{victim}| < \frac{r}{2^{N-1}}.$$

Similarly,

$$|y_{est} - y_{victim}| < \frac{r}{2^{N-1}}. \tag{7}$$

The final localization error between the target actual position and the estimated position is

$$\begin{aligned} err &= \sqrt{(x - x_{victim})^2 + (y - y_{victim})^2} \\ &< \frac{\sqrt{2}r}{2^{N-1}}. \end{aligned} \tag{8}$$

Thus, our method can achieve any small localization accuracy with more queries.

5 Experiments

In the sections before, we discussed that our algorithm can reach any precision in theory. However, there are always some differences between the actual distance and the measured distance. Nowadays the popular locating ways of mobile phone include GPS, Network Locating and the blending of the both. There will be inevitable errors regardless of locating ways. At the same time, the application would like to add errors to the location information to protect the privacy of the users. As a result, it is important for locating model to be robust when there exist errors that can not be negligible. Here we simulate our model with different error distribution settings and different band distance r.

5.1 Model Settings

The error model settings refer to [10]. Considering that errors exist even for short distance, we change the error to non-zero when the actual distance is less than 100 m for different models.

$$err = \begin{cases} exprnd(1), Exponential; \\ unirnd(0,5), Uniform; \\ raylrnd(\frac{1}{1.253}), Rayleigh; \\ normrnd(1, \frac{1}{3}), Gaussian. \end{cases} \tag{9}$$

In our simulation, we set the error tolerance $\epsilon = 1$ and suppose that the first measurement (i.e., $|A_0V|$ in Fig. 1) is accurate. The adding error is $e\hat{r}r$ in the simulation which satisfies $e\hat{r}r = n \cdot err, n \in Z^+$ (error is generated from the above model).

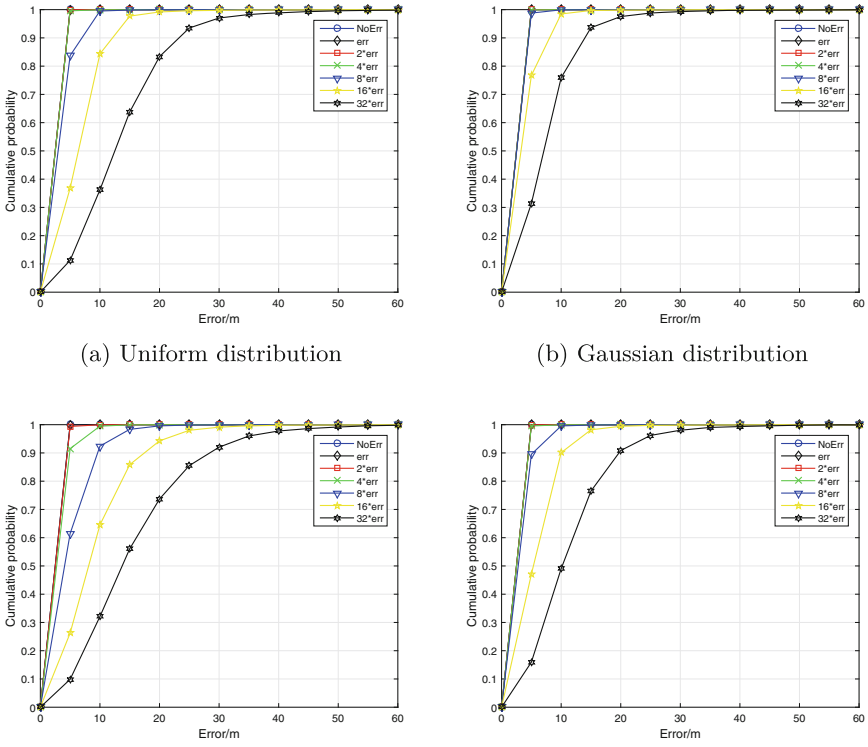
5.2 Simulation

Effects of error distribution models. Assume that $r = 100$ m, our model only leverage the information of small distances. When the reporting distance is bigger than $2r$, it is useless. To exploit the model robustness well, we enlarge the original errors generated by above settings by timing n, and $n \in \{1, 2, 4, 8, 16, 32\}$ (i.e.,$e\hat{r}r = n \cdot err$).

Figure 3 shows the localization error distribution under different models with different mean. We could find that our model works well even if the error is very big (such as $e\hat{r}r = 32err$). The worst case whose incorporating error distribution is exponential demonstrates that the localization error cumulative probability can reach 70% when the corresponding error is within 20 m. The remain cases increase approximately 10% higher, and the model works best for Gaussian distributions.

Effects of band distance r. We set $e\hat{r}r = err$, $r \in \{100\,\text{m}, 200\,\text{m}, 400\,\text{m}, 800\,\text{m}\}$. Figure 4 shows the result with different r. We find that the performance is worst when incorporating exponential distribution error. When r is small like less than 400 m, the localization error cumulative probability is close to 100% within 20 m for exponent distribution and close to 100% within 10 m for other error models; even r is bigger such as $r = 800$ m, the cumulative probability is close to 80% within 40 m for exponential distribution and close to 100% within 40 m for other error models. Still the model performs best for gaussian distribution.

The above results show our localization methodology can efficiently get rid of the errors. Taking the locating details into consideration, we should know that the localization precision is decided by the refinement step *LocateToR* of our model. The following reasons may lead to our algorithm robustness: (i) In *LocateAccurate*, we locate the victim by adding constraints to the horizontal



(a) Uniform distribution (b) Gaussian distribution

(c) Exponential distribution (d) Rayleigh distribution

Fig. 3. Localization error distributions under various error models with different average value settings

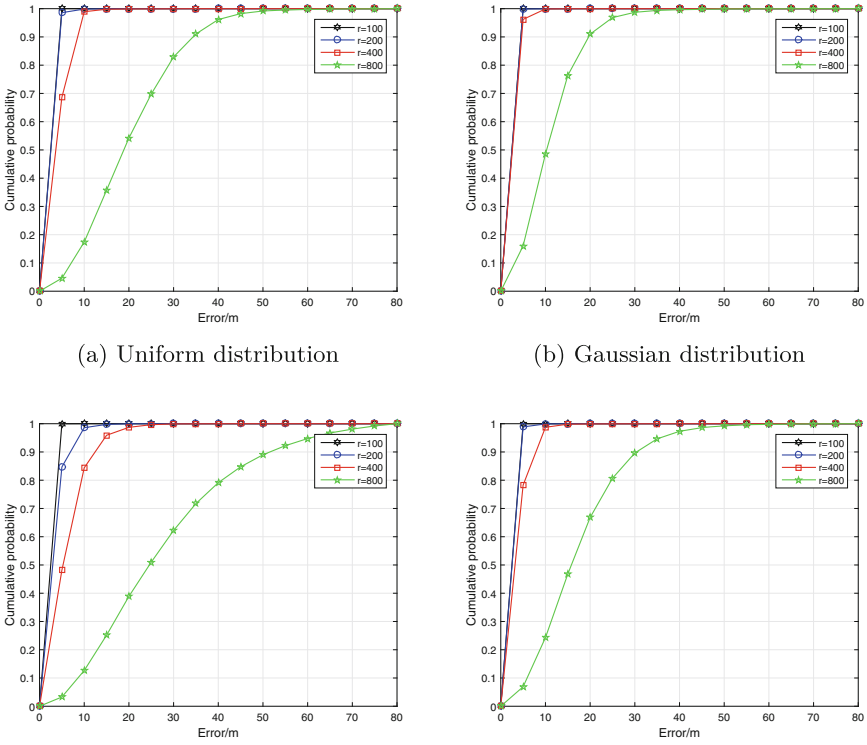
coordinates and vertical coordinates then estimate the position by taking average of the constraint points. Taking average may cancel the errors because all the errors are positive. (ii) When determining the constraint points, the movement step size is determined. And the resulting error must be caused by the previous wrong movement. As our algorithm is iterative, the previous error can be corrected by the following movement.

The model performs best when incorporating Gaussian error models. We conjecture that Gaussian distributions are symmetric distributions and the error may be cancelled by taking the average.

6 Discussions

6.1 Complexity Analysis

In this section, we analyze the query complexity of the proposed localization algorithm. It consists of the two parts: Complexity of $LocateToR$ and Complexity of $LocateAccurate$. The first part is determined by the θ in Fig. 1.



(a) Uniform distribution (b) Gaussian distribution

(c) Exponent distribution (d) Rayleigh distribution

Fig. 4. Localization error distribution with different r for different error models

$$\begin{aligned} \Omega_1 &= \frac{1}{2} \cdot \frac{2\pi}{2\theta} \\ &= \frac{\pi}{2arccos\left(\frac{(d^2+(d-\frac{r}{2})^2-r^2)}{2d(d-\frac{r}{2})}\right)}. \end{aligned} \tag{10}$$

The second part is decided by the tolerance ϵ, the smaller ϵ is, the more queries will be required.

$$\begin{aligned} \Omega_2 &= 4N \\ &= 4 \cdot \left(\left\lceil log_2 \frac{\sqrt{2}r}{\epsilon} \right\rceil + 1\right). \end{aligned} \tag{11}$$

As a result, the query complexity of the entire model is

$$\begin{aligned} \Omega &= \Omega_1 + \Omega_2 \\ &= \frac{\pi}{2arccos\left(\frac{(d^2+(d-\frac{r}{2})^2-r^2)}{2d(d-\frac{r}{2})}\right)} + 4 \cdot \left(\left\lceil \log_2 \frac{\sqrt{2}r}{\epsilon} \right\rceil + 1\right). \end{aligned} \tag{12}$$

6.2 Comparison Remarks

The recent research progress of the privacy problems have driven popular LBSN service providers to enhance their apps to track more user abnormal behaviors for privacy protection. It has become more difficult to conduct experiments in real-world systems. Nevertheless, we believe that the battle between location attacks and protection will be still continuing. In this subsection, we compare this paper with a few representative studies. [7,9,10] focus more on using a number of probes in order to decrease the obfuscation. These methods require significant probe cost while the localization results are less accurate than this work. [4] proposed a similar spatial iteration attack approach yet with $3 - 5$ s response time. Our method is less time-consuming. [17] requires to train a supervised or unsupervised model. The results are promising yet the data-collection procedure is expensive. In summary, our two-step method provides a practical attack method with high accuracy and small time complexity.

7 Conclusion

Mobile app people-nearby services have often been providing band distances instead of exact distances in order to protect privacy. In this paper, we revisit localization attacks in band-based distances. We proposed a new model to launch localization attacks and proved that our method can pinpoint target users accurately with the theoretically settings. When we incorporated different error distributions into our model, the simulation results showed that out model can efficiently combat against the impact the errors. This proposed model is robust with all most all band distances. In this paper, we continuously investigate the privacy leakage problem from end-systems. In the emerging software defined wireless networks, the network infrastructure may provide more data functions [18]. We envision that the privacy leakage problem may become more severe in the coming software defined edge computing and networking era.

Acknowledgments. This work was supported in part by the National Natural Science Foundation of China (No. 61370231), and in part by the Fundamental Research Funds for the Central Universities (No. HUST:2016YXMS303).

References

1. Noack, R.: Could using gay dating app Grindr get you arrested in Egypt? The Washington Post, 12 September 2014
2. Paton, C.: Grindr urges LGBT community to hide their identities as Egypt persecutes nation's gay community. The Independent, 26 September 2014
3. Li, M., Zhu, H., Gao, Z., Chen, S., Yu, L., Hu, S., Ren, K.: All your location are belong to us: breaking mobile social networks for automated user location tracking. In: 15th ACM International Symposium on Mobile Ad Hoc Networking and Computing, pp. 43–52 (2014)

4. Polakis, I., Argyros, G., Petsios, T., Sivakorn, S., Keromytis, A.D.: Where's wally?: Precise user discovery attacks in location proximity services. In: ACM SIGSAC CCS, pp. 817–828 (2015)
5. Wang, G., Wang, B., Wang, T., Nika, A., Zheng, H., Zhao, B.Y.: Whispers in the dark: analysis of an anonymous social network. In: ACM Internet Measurement Conference, pp. 137–150 (2014)
6. Ding, Y., Peddinti, S.T., Ross, K.W.: Stalking Beijing from Timbuktu: a generic measurement approach for exploiting location-based social discovery. In: ACM Workshop on Security and Privacy in Smartphones & Mobile Devices (2014)
7. Xue, M., Liu, Y., Ross, K.W., Qian, H.: I know where you are: thwarting privacy protection in location-based social discovery services. In: IEEE Conference on Computer Communications Workshops (2015)
8. Xue, M., Liu, Y., Ross, K., Qian, H.: Thwarting location privacy protection in location-based social discovery services. Secur. Commun. Netw. **9**(11), 1496–1508 (2016)
9. Peng, J., Meng, Y., Xue, M., Hei, X., Ross, K.W.: Attacks and defenses in location-based social networks: a heuristic number theory approach. In: International Symposium on Security and Privacy in Social Networks and Big Data (SocialSec), pp. 64–71 (2015)
10. Cheng, H., Mao, S., Xue, M., Hei, X.: On the impact of location errors on localization attacks in location-based social network services. In: Wang, G., Ray, I., Alcaraz Calero, J.M., Thampi, S.M. (eds.) SpaCCS 2016. LNCS, vol. 10066, pp. 343–357. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-49148-6_29
11. Liu, J., Zhang, Y., Zhao, F.: Robust distributed node localization with error management. In: Proceedings of the 7th ACM International Symposium on Mobile Ad Hoc Networking and Computing, pp. 250–261 (2006)
12. Qin, G., Patsakis, C., Bouroche, M.: Playing hide and seek with mobile dating applications. In: Cuppens-Boulahia, N., Cuppens, F., Jajodia, S., Abou El Kalam, A., Sans, T. (eds.) SEC 2014. IAICT, vol. 428, pp. 185–196. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-642-55415-5_15
13. Mascetti, S., Bertolaja, L., Bettini, C.: A practical location privacy attack in proximity services. In: IEEE 14th International Conference on Mobile Data Management (MDM), vol. 1, pp. 87–96 (2013)
14. Correa, D., Silva, L.A., Mondal, M., Benevenuto, F., Gummadi, K.P.: The many shades of anonymity: characterizing anonymous social media content. In: International AAAI Conference on Web and Social Media (2015)
15. Xue, M., Yang, L., Ross, K.W., Qian, H.: Characterizing user behaviors in location-based find-and-flirt services: anonymity and demographics. Peer-to-Peer Netw. Appl. **10**(2), 357–367 (2017)
16. Wang, R., Xue, M., Liu, K., Qian, H.: Data-driven privacy analytics: a WeChat case study in location-based social networks. In: Xu, K., Zhu, H. (eds.) WASA 2015. LNCS, vol. 9204, pp. 561–570. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-21837-3_55
17. Xue, M., Ballard, C., Liu, K., Nemelka, C., Wu, Y., Ross, K., Qian, H.: You can yak but you can't hide: localizing anonymous social network users. In: ACM IMC, pp. 25–31 (2016)
18. Chen, Z., Fu, D., Gao, Y., Hei, X.: Performance evaluation for software defined WiFi DCF networks from theory to testbed. In: 16th IEEE International Conference on Ubiquitous Computing and Communications (IUCC) (2017)

http://www.springer.com/978-3-319-72388-4

Security, Privacy, and Anonymity in Computation, Communication, and Storage
10th International Conference, SpaCCS 2017, Guangzhou, China, December 12-15, 2017, Proceedings
Wang, G.; Atiquzzaman, M.; Yan, Z.; Choo, K.-K.R. (Eds.)
2017, XVIII, 610 p. 200 illus., Softcover
ISBN: 978-3-319-72388-4