POSTER: Positioning Attack on Proximity-Based People Discovery

Huan Feng
The University of Michigan
Michigan, USA
huanfeng@umich.edu

Kang G. Shin
The University of Michigan
Michigan, USA
kgshin@umich.edu

ABSTRACT

Over the past few years, *Proximity-based People Discovery* (PBPD) services, typically known as *Nearby Friends*, have been increasingly popular among geosocial apps. Unlike many unsuccessful predecessors which directly pinpoint users' exact locations on the map, PBPD services provide coarse-grained (discretized) proximity information, such as "Jennifer is within 2 miles," striking a useful balance between privacy and functionality. Considering PBPD's business potential, many companies including Facebook have been trying to promote this feature and instill the perception in mobile users that coarse-grained proximity information is innocuous to share. Here, we propose a novel positioning attack which can locate end-users of PBPD services with high precision using only coarse-grained (discretized) proximity information. This attack requires neither specialized hardware nor server-side collusion and can be easily automated. Based on this attack, we design and implement *Geosocial Positioning System* (GsPS) and show that GsPS can effectively locate users with high precision ($10m$) in a matter of a few minutes under real-world settings, and is capable of performing effective city-scale scanning and long-term profiling at low costs. The public and the social network industry should therefore be aware of the potential risk introduced by this attack and consider use of PBPD services with caution.

Categories and Subject Descriptors

C.2.0 [**Computer-Communication Networks**]: General—*Security and protection*; K.4.1 [**Computers and Society**]: Public policy issues—*privacy*

Keywords

Geosocial network, location privacy, triangulation

1. INTRODUCTION

Proximity-Based People Discovery (PBPD) has become popular in geosocial apps over the past few years. It allows users to discover, learn and interact with others nearby, providing a natural link between online social activities and the physical world. By sharing his exact location with the service provider, a user gets prompted with a list of nearby users ranked by geographical proximity. Although the service provider knows the exact location of every user, it only discloses coarse-grained proximity information such as "Jenifer is within 2 miles." Compared to its predecessor which directly pinpoints users' exact locations on the map, this proximity-based model is believed/perceived to be much safer in terms of location privacy, thus getting widely adopted by many popular apps, such as Facebook.

However, we show this perceived safety to be false by presenting an effective positioning attack. This attack demonstrates that coarse-grained proximity information — which is generally believed to be innocuous — can be exploited to infer users' locations with high precision (within 10m) in a matter of a few minutes. The proposed attack adapts the well-known triangulation positioning technique to the unique properties of geosocial apps. By measuring the target's distance to a series of strategically selected reference locations (users), the target's candidate area can be shrunk effectively. Based on this positioning attack, we design and implement *Geosocial Positioning System* (GsPS) which requires neither specialized hardware nor server-side collusion, and can be easily automated to establish city-scale scanning and long-term profiling at negligible costs. We test GsPS on two representative real-world apps with PBPD services: Facebook and Grindr. Our experimental results show that GsPS is very effective in practical settings.

2. ATTACK OVERVIEW

2.1 Threat Model

The proposed attack allows an adversary to precisely locate other users using the same PBPD service. A victim can be any ordinary user whose proximity information is visible to the attacker. A prior trust relationship may be required between the attacker and the victim if the geosocial app discloses proximity information only between friends. The threat of this attack depends on how each app balances three aspects: (1) the granularity of proximity information, (2) who can access proximity information, and (3) the frequency of updating proximity information. For example, a service that allows a stranger to continuously track your exact location is definitely more disturbing (and riskier) than the one that only discloses coarse-grained proximity information to your friends when you explicitly checks in.

CCS'14, November 3–7, 2014, Scottsdale, Arizona, USA.
ACM 978-1-4503-2957-6/14/11.
http://dx.doi.org/10.1145/2660267.2662389.

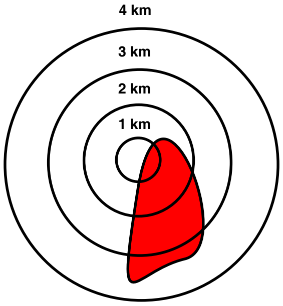


Figure 1: Each scan partitions the candidate area into a group of concentric rings and keeps one of them.

2.2 Basic Design

Triangulation is one of the most commonly-used techniques for locating an object. By precisely measuring the target's distances to three (or more) references, its exact location can be computed by intersecting the circles (or spheres) centered at each reference with the measured distance as the radius. However, applying the basic idea of triangulation to PBPD is challenging because it takes proximity information of various discretized formats which give very different implications on the actual location of a user. With direct application of the traditional triangulation algorithm on PBPD services, an attacker can only get a rough candidate area which typically spans hundreds of meters or even several kilometers.

To meet this challenge, we design an iterative positioning algorithm which can infer the user's exact location effectively with high precision, even with discretized proximity information. Our algorithm reduces the candidate area a user may reside in by scanning at a series of strategically selected reference locations. As Fig. 1 shows, each scan partitions the candidate area into a group of concentric rings and keeps one of them. We show that this reduction can be very effective if the reference locations are carefully chosen. Our algorithm selects each reference location according to the intuition that the expected area of the candidate space should be minimized after scanning at this location.

We evaluate the effectiveness of our positioning algorithm by measuring the number of steps (scans) taken to achieve a pre-specified positioning precision. The less scans an algorithm requires, the less time and resources are consumed, and the less suspicious the attacker will appear. By precision we mean the distance between the target's actual and estimated locations. We adopt the proximity format used by Facebook app in our simulation: the proximity information is discretized to mile granularity when larger than one mile, and to half mile when smaller than one mile. We measured the number of steps taken to achieve a precision of $10m$, with the attacker initialized at 1 mile, 5 miles and 10 miles away from the target. The experiments are repeated 100 times under each setting and the corresponding CDFs (Cumulative Distribution Functions) are plotted in Fig. 2. The simulation results indicate that to locate the target with high precision (e.g., $10m$), 10–20 scans would suffice. This number demonstrates that our positioning algorithm is effective enough to support real-world attacks, but a certain level of automation is required (or at least preferred).

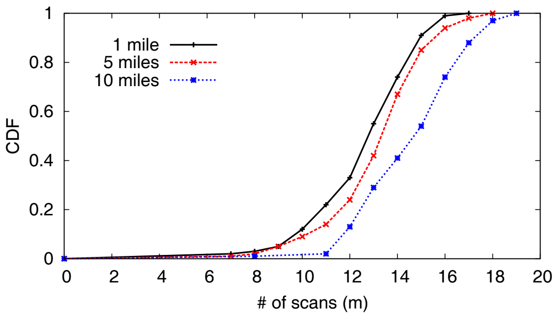


Figure 2: Cumulative distribution function (CDF) of the number of scans required to achieve $10m$ precision.

2.3 GsPS: Geosocial Positioning System

GsPS consists of four components: (1) a standalone attack engine written in Java which provides universal algorithmic support for the triangulation attack, (2) Android emulator(s) installed with mock location provider and customized plug-ins for various PBPD apps, (3) server-side encapsulation (called *Drone*) of the Android emulator which bridges the attack engine and the Android emulators, and (4) a GUI (Graphical User Interface) for debugging and testing. Fig. 3 depicts an overview of the design.

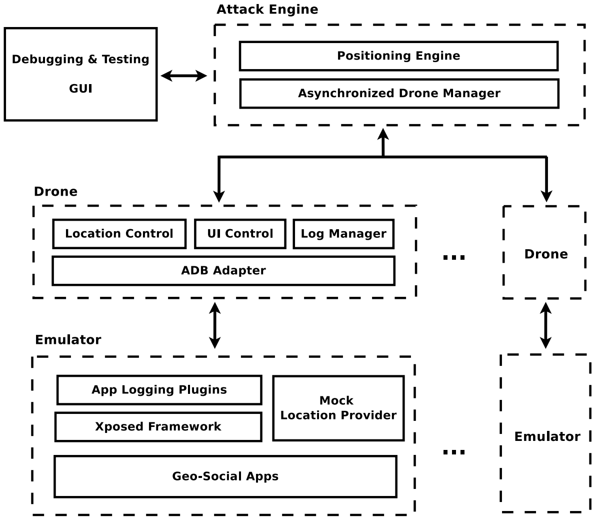
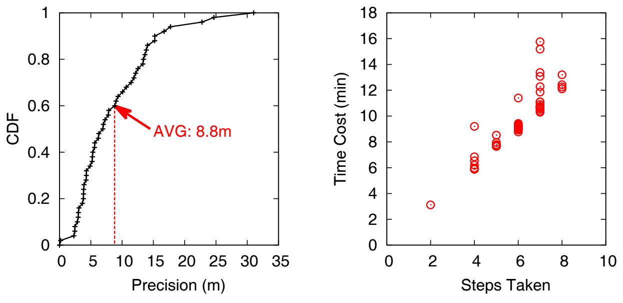


Figure 3: An overview of the design of GsPS

The app logging plug-ins automatically extract the proximity information by intercepting the interactions between the app and the Android framework. This information is then synced with the attack engine using the *Drone* interface. Each *Drone* instance communicates with the corresponding emulator via Android Debugging Bridge (ADB). An attacker can distribute the workloads among multiple emulators either to speed up the positioning process or to achieve a higher throughput.

3. EXPERIMENTAL RESULTS

We host GsPS on a workstation with four Intel Xeon(R) CPUs (3.2GHz), 16GB RAM and configure it to terminate when the max error (worst possible precision) reduces to less than $30m$. Here, we test GsPS under three representative scenarios using real-world apps.



(a)

(b)

Figure 4: (a) The cumulative distribution function (CDF) of the attack precision on Grindr; (b) Scatter plot of the number of scans taken and the time cost

Attack Scenario 1: *An attacker wants to locate some random stranger he's interested in when he browses an app providing PBPD service among strangers.*

We test this attack scenario on Grindr, a geo-social app geared towards gay, bisexual, and bi-curious men and has a daily active user over one million in 192 countries. The exact location of each Grindr user is very sensitive given the property of the user group of this app. The proximity information Grindr provides is continuous when less than $1km$, and discretized to km when larger than $1km$. We registered two accounts on Grindr — one as the target and the other as the attacker — and mount our positioning attack with GsPS. The attacker is initialized at some random location within $30km$ radius of the target. Fig. 4 shows the results (50 trials) of our positioning attack on Grindr.

Attack Scenario 2: *Instead of locating some specific user, a dedicated attacker wants to locate all Grindr users within a specific area.*

We launch a scalable attack by initializing 5 emulators at some random locations in the San Francisco area. These emulators independently scan and locate their nearby friends and merge the results, i.e., estimated locations of the targets. As Fig. 5 shows, GsPS located 60 different users in less than 10 minutes and the effectiveness of different emulators can vary greatly. In general, the attack is more effective when scanning in the crowded area, such as downtown, and less effective in a suburban area where users are scattered widely.

Attack Scenario 3: *An attacker who's interested in a specific user tries to profile him over time and extract his Points of Interests (PoIs).*

Facebook released a proximity-based Nearby Friends feature recently (in April 2014) which notifies users in real time when their friends are nearby. This type of real-time feature allows an attacker to build a timestamped view of a user's locations during a certain time period and thus poses profiling threats. Here, we fed a Facebook user a 10-hour location trace of a real user in San Francisco and locate him using GsPS. We find that GsPS can effectively identify the PoIs (location clusters in Fig. 6) but may miss the location points when the user is constantly moving.

4. CONCLUSION

Of all variants of Location-Based People Discovery services (LBPDs), the proximity-based ones are the most widely deployed for their natural preservation of privacy. Our positioning attack, however, shakes the foundation of PBPD features, showing that they are not any safer than directly disclosing a user's exact location to others. This attack assumes neither specialized hardware, nor server-side collusion, and can be easily automated. This new attack vector poses serious threats to the social network industry, and also draws attention from the research community [2]. Although numerous location protection mechanisms have been proposed for friend discovery services in geosocial networks [3, 1], it remains unclear whether they can achieve reasonable trade-offs between usability, performance and privacy on this new attack. Therefore, there is an urgent need for rethinking proximity-based social features, given all potential risks introduced by this attack.

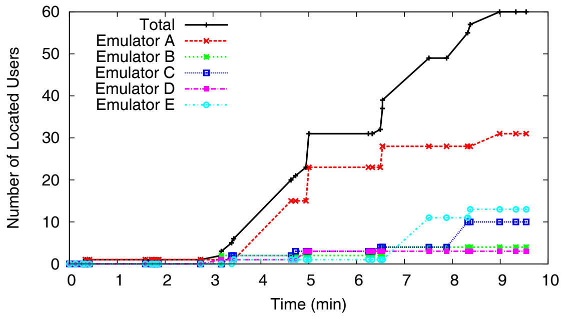


Figure 5: 10-min scalable attack on Grindr in the San Francisco area using five emulators.

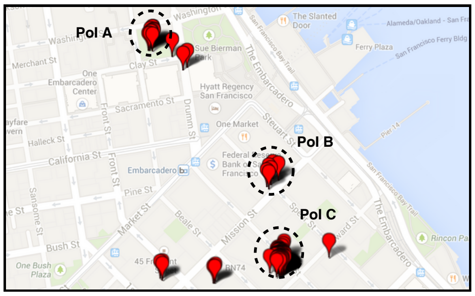


Figure 6: 10-hour profiling attack on a Facebook user to extract his Points of Interests (PoIs).

The work reported here is supported in part by the US NSF under Grant CNS-1114837.

5. REFERENCES

[1] H. P. Li, H. Hu, and J. Xu. Nearby friend alert: Location anonymity in mobile geosocial networks. *IEEE Pervasive Computing*, 12(4):62–70, Oct. 2013.

[2] M. Li, H. Zhu, Z. Gao, S. Chen, L. Yu, S. Hu, and K. Ren. All your location are belong to us: Breaking mobile social networks for automated user location tracking. In *Proceedings of the 15th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, MobiHoc '14, pages 43–52, New York, NY, USA, 2014. ACM.

[3] S. Mascetti, D. Freni, C. Bettini, X. S. Wang, and S. Jajodia. Privacy in geo-social networks: Proximity notification with untrusted service providers and curious buddies. *The VLDB Journal*, 20(4):541–566, Aug. 2011.