Breaking Mobile Social Networks for Automated User Location Tracking

Muyuan Li, Haojin Zhu, Zhaoyu Gao, Si Chen, Le Yu, Shangqian Hu, and Kui Ren

ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc), 2014

Outline

- Location Information Management in Mobile Social Networks
- Our Automated Attack Framework for User Location Discovery
- Our Real-world Attack Experiment and Results
- The Proposed Defense Mechanisms
- Concluding Remarks

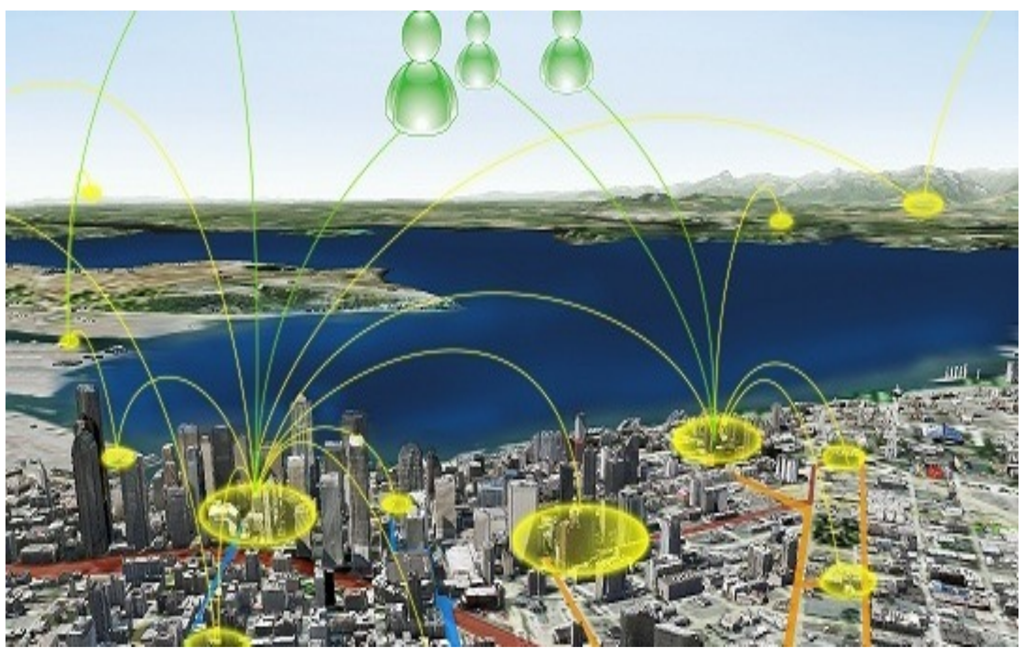
Outline

- Location Information Management in Mobile Social Networks
- Our Automated Attack Framework for User Location Discovery
- Our Real-world Attack Experiment and Results
- The Proposed Defense Mechanisms
- Concluding Remarks

Mobile Social Networks

- The ubiquity of the smartphones has led to the extreme popularity of mobile social networks
- Billions of users are actively using them for social interactions on a daily basis
- Successful examples include Wechat, Facebook, Google+, What’s App, Momo, etc.

Location Based Social Networks (LBSNs)



- Location information is key to user interaction experiences in mobile social networks today
- They are used to enable and facilitate various location-based social interactions

Examples of Location Based Social Interactions

- Checking-in Services
  - Allow users to check in to report their locations: Facebook, Weibo, Foursquare, etc.
- Geotagging
  - Reveal/Redact location data on user posts/messages: Facebook, Weibo, Renren, etc.
- Location-dependent Comments
  - Comment to specific subjects with exact known locations: Yelp, Dianping, etc.
- And most popularly, proximity-based friend discovery
  - The focus of this research

How do LBSNs Acquire Users' Location?

- Mobile users voluntarily report their location via LBSN client App to LBSN servers.
- Various types of location information. i.e., Wi-Fi (80m), GPS (10m) and Cell ID (600m), are being collected in a periodical, on-movement or combined fashion.

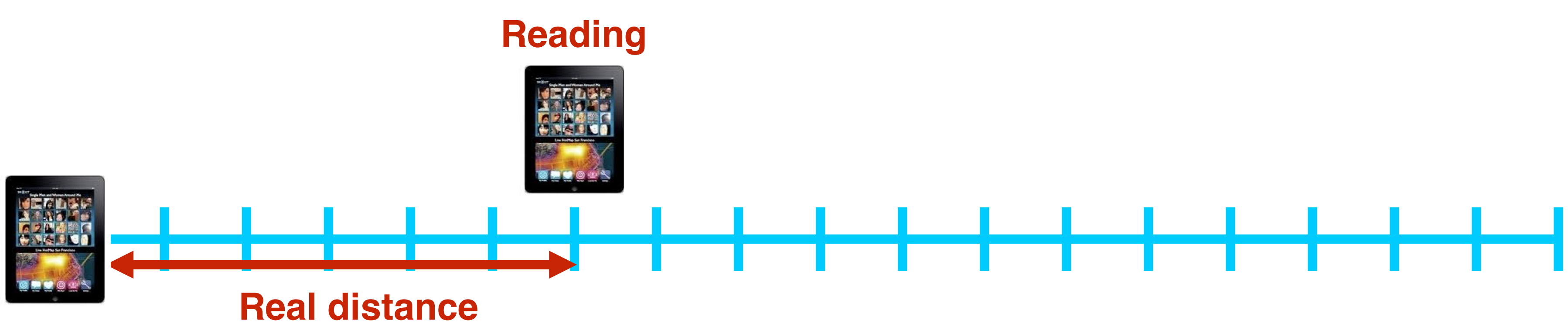
LBSN	Location Retrieval Method
Momo	Rely on Baidu location SDK to fuse inputs from multiple location sources
Wechat/Skout	Select the available one with the highest precision (GPS, Wi-Fi, Cell ID)

How Accurate are the Displayed Location Info?

- The location accuracy displayed in LBSN client App's is reflected by the distance between
  - an user's real physical location (as perceived by the location services according to his/her mobile devices) and
  - the location readings from LBSN App's.
- Location accuracy varies across different LBSNs
  - Depending on both their own internal processing strategy and
  - The available location info sources
- The overall observation is that LBSN location readings are quite accurate.

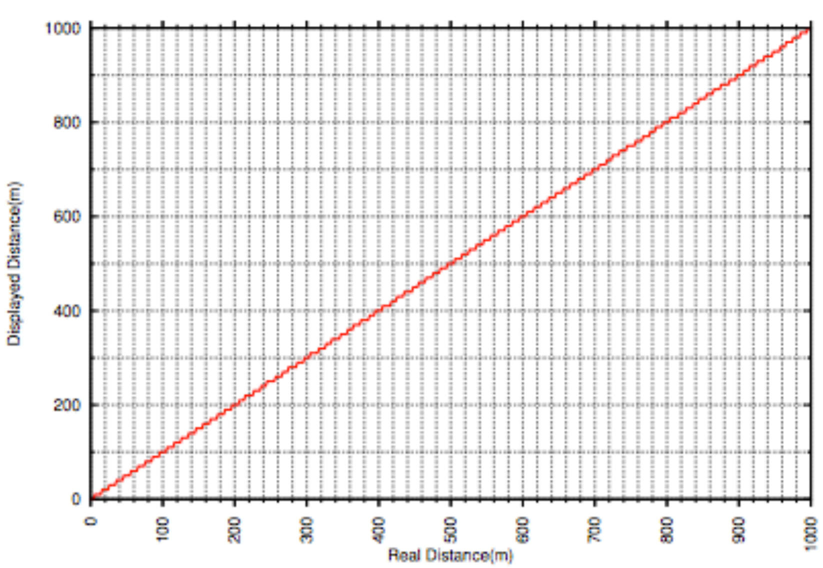
Location Accuracy Testing in Popular LBSNs

- Differences between the real distances and the distance readings on popular LBSNs are studied:
  - Run two instances of the same LBSN in two VMs;
  - Fix one reference point in one instance and move a testing point along a line in the other;
  - Record the actual distances between the testing point and the reference point;
  - Refresh the LBSN App in the VMs and record their distance readings

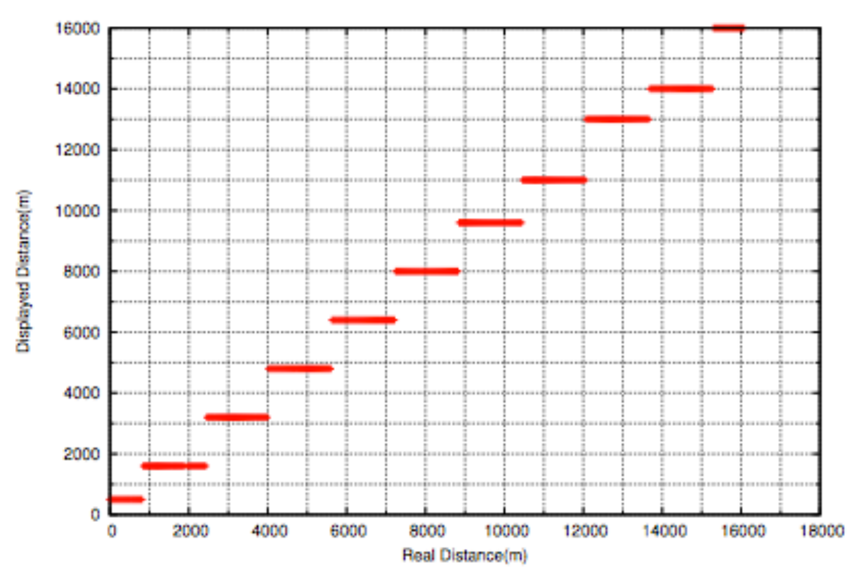


Location Accuracy Testing Results

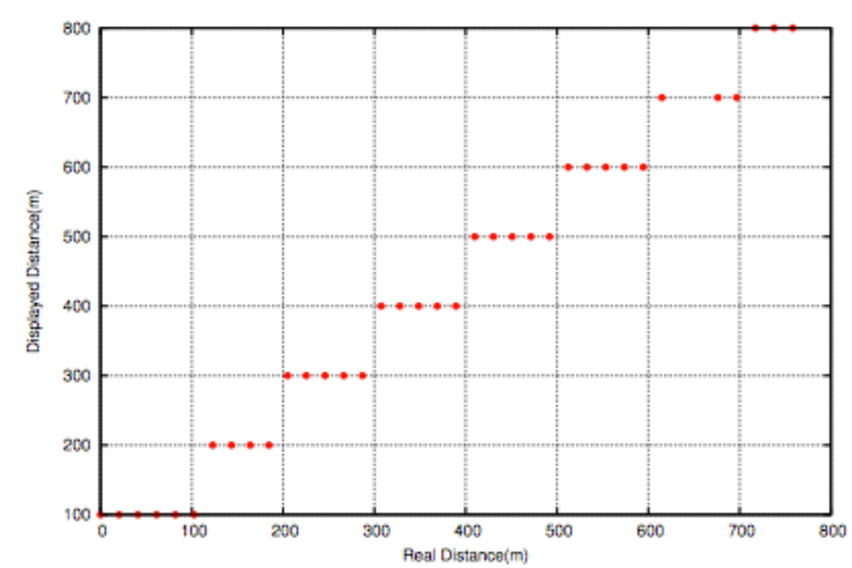
- Momo's distance readings are rounded at 5m;
- Skout rounds up the distance readings every 1.0mi but also indicates when a user is within 0.5mi;
- Wechat answers user's location with the precision of 100m when the distance <1km in metropolitan areas;



Momo



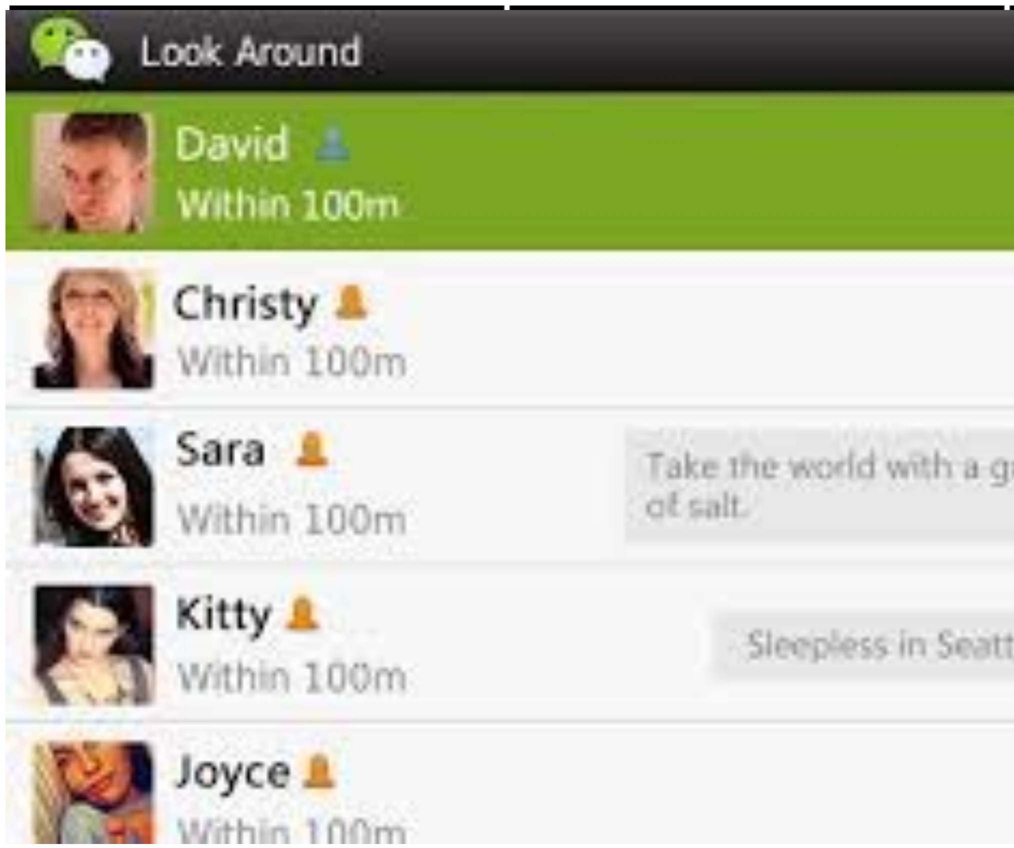
Skout



Wechat

How are the Location Info Shared among LBSN Users?

- Open direct access to any registered users
  - Sharing exact locations among users
- Authorized direct access
  - Sharing locations with authorized friends
- Indirect access with constraints
  - Sharing obfuscated location information according to various constraints

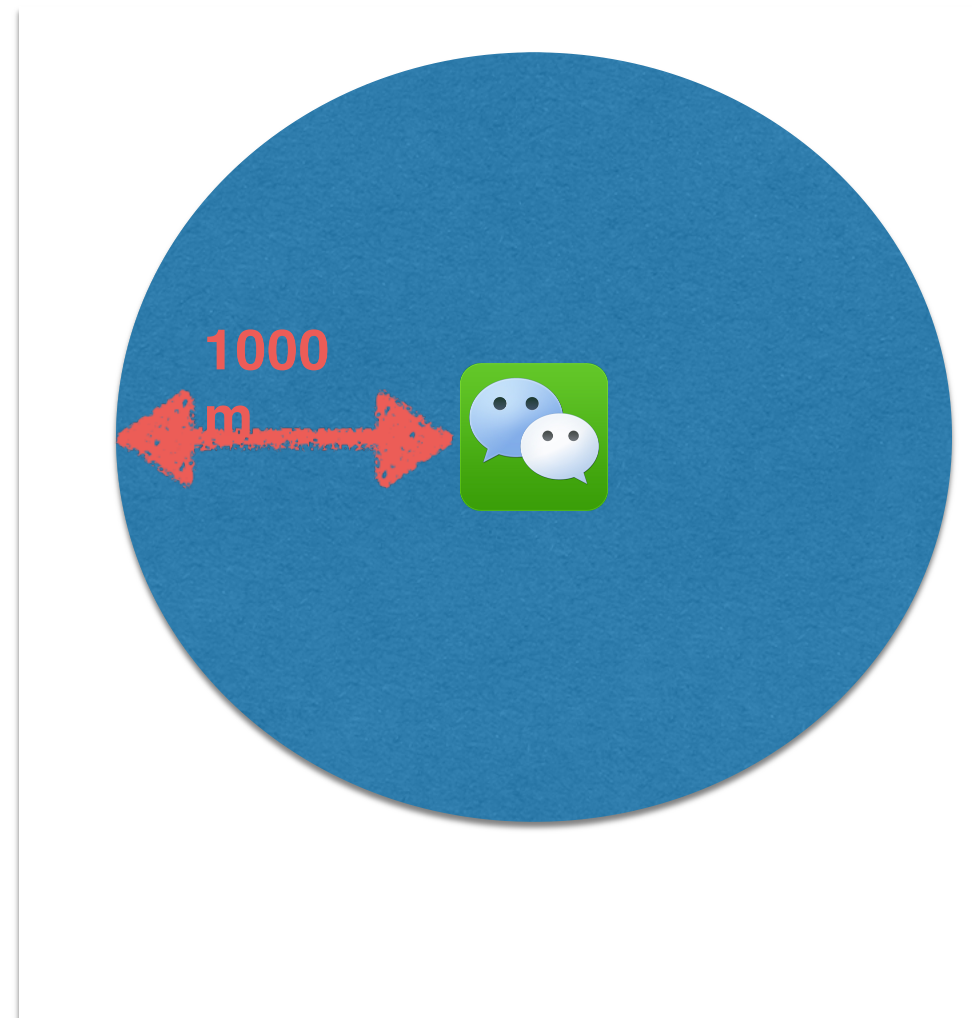


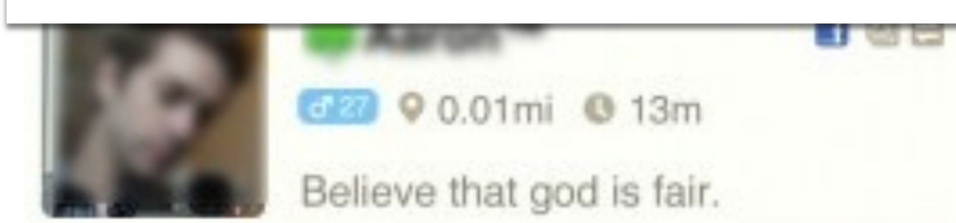
An Overlook of Location Sharing in Popular LBSNs

	Number of users	Classification
Wechat	300 millions	Indirect
Skout	5 millions	Indirect
Momo	30 millions	Indirect
iAround	10 millions	Open direct access
Google+	30 millions	Authorized direct access
Facebook	1.23 billion	Authorized direct access

A Closer Look on Constrained Indirect Access in LBSNs

- Showing the relative distance between users
  - e.g., Momo displays relative distances with the precision of 5m
  - Users see their distances to other users instead of exact location.
- Imposing a minimal location accuracy constraint
  - e.g., Skout shows relative distances no smaller than 0.5 mi;
  - Users see their distances to other users when the distances are larger than 0.5mi with with the precision of 1mi.
- Imposing a maximum distance constraint
  - e.g., Wechat lists only the users within the range of 1km in metropolitan areas with the precision of 100m;
  - Users cannot see others who are more than 1km away.





Location Privacy in Existing LBSNs

- Today, user location privacy achieved relying on indirect & constrained location sharing; Exact location info never shared among users
  - Such utility and privacy trade-offs are today's industry best practices, affecting hundreds of millions of users.
  - Viewed by most popular LBSNs as a desirable middle ground to both protect user location privacy and enable effective location-based services

Outline

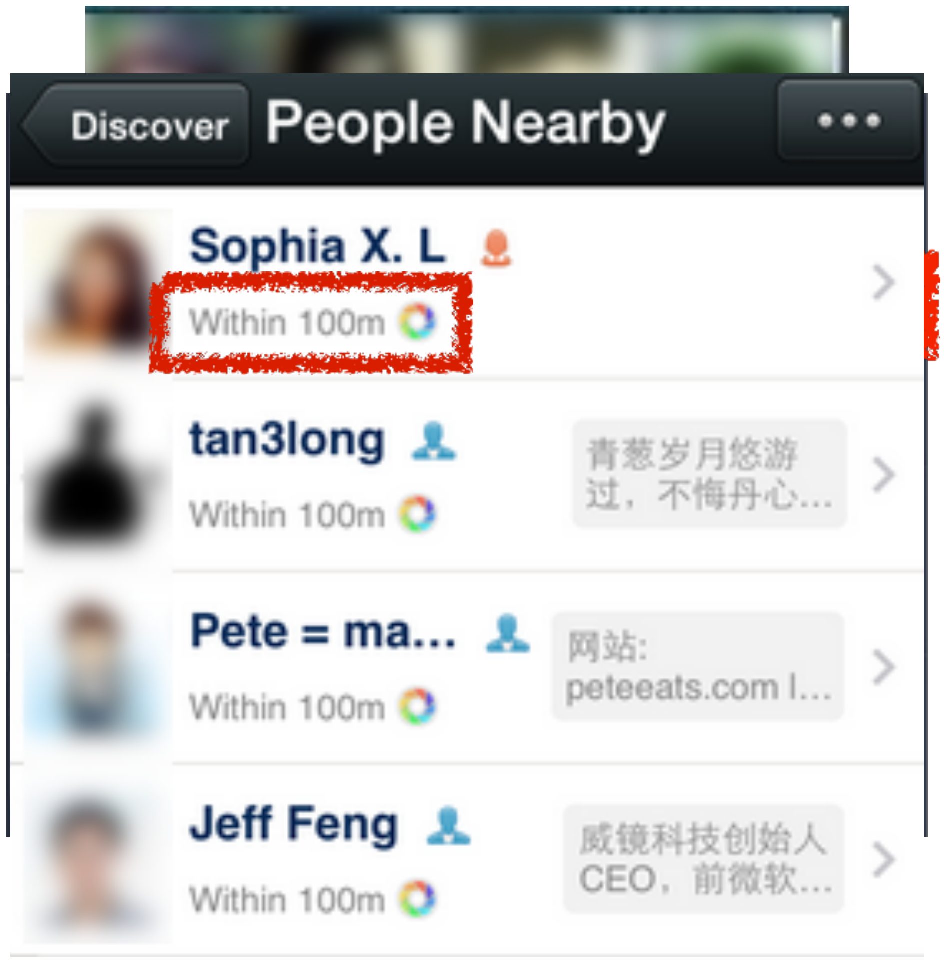
- Location Information Management in Mobile Social Networks
- Our Automated Attack Framework for User Location Discovery
- Our Real-world Attack Experiment and Results
- The Proposed Defense Mechanisms
- Concluding Remarks

The Attack Goal & Assumptions

- We assume only a weak outsider adversary:
  - having exactly the same privilege as ordinary users
  - exploiting only publicly available information without hacking into the LBSN servers
- Yet, such adversary is still able to achieve both:
  - localizing an arbitrary user with very high accuracy
  - performing long-term tracking and eventually revealing the user's identity information with high probability

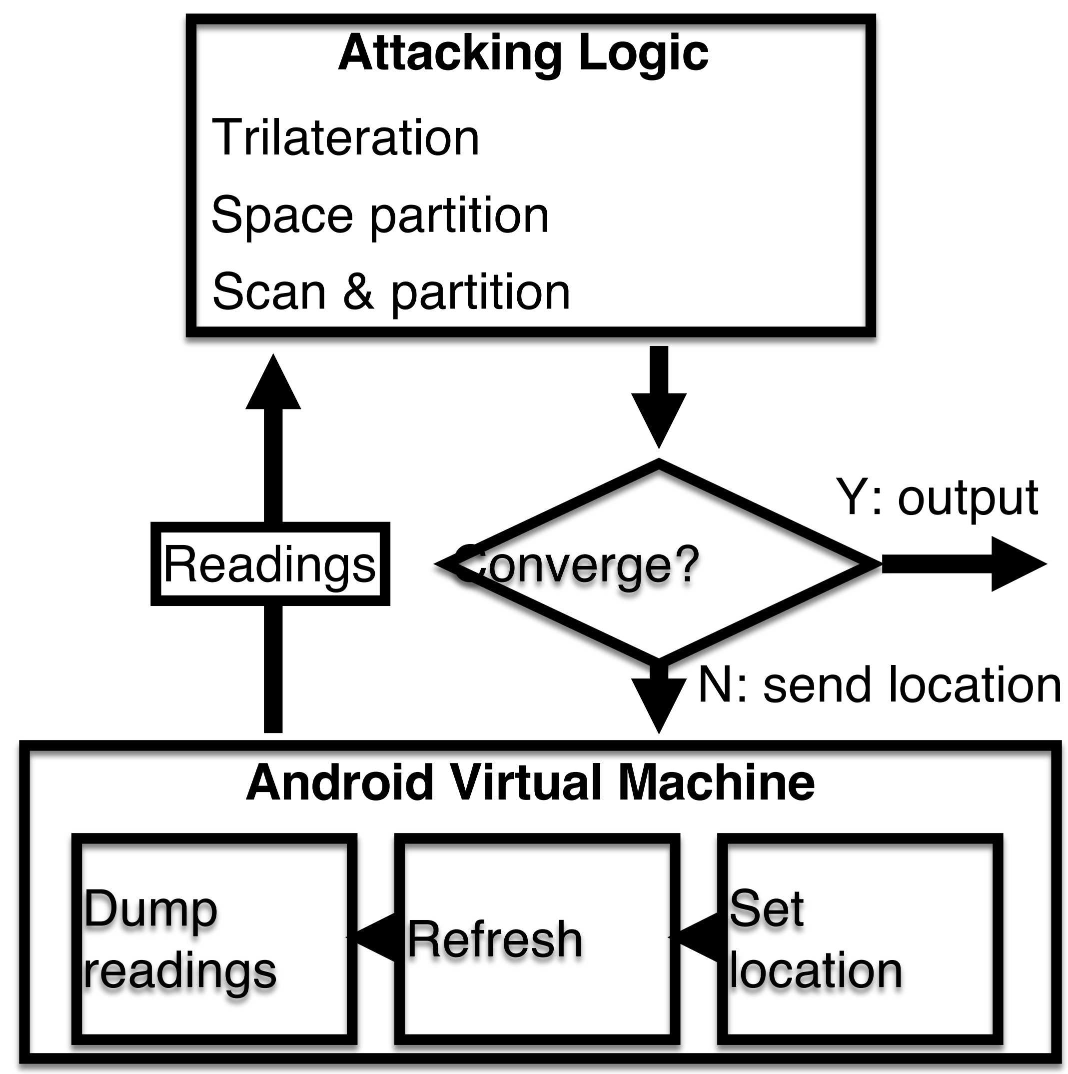
Victim Targeting

- The LBSNs provide query interfaces to retrieve proximity of an arbitrary user without raising the victim's attention:
  - In Momo, proximity can be read when the attacker searches the victim by User ID
  - In Skout, the attacker sends a regular message to the victim and the proximity will be displayed for following queries
  - In Wechat, the attacker searches "People Nearby" and the proximity is shown along with the victim's ID



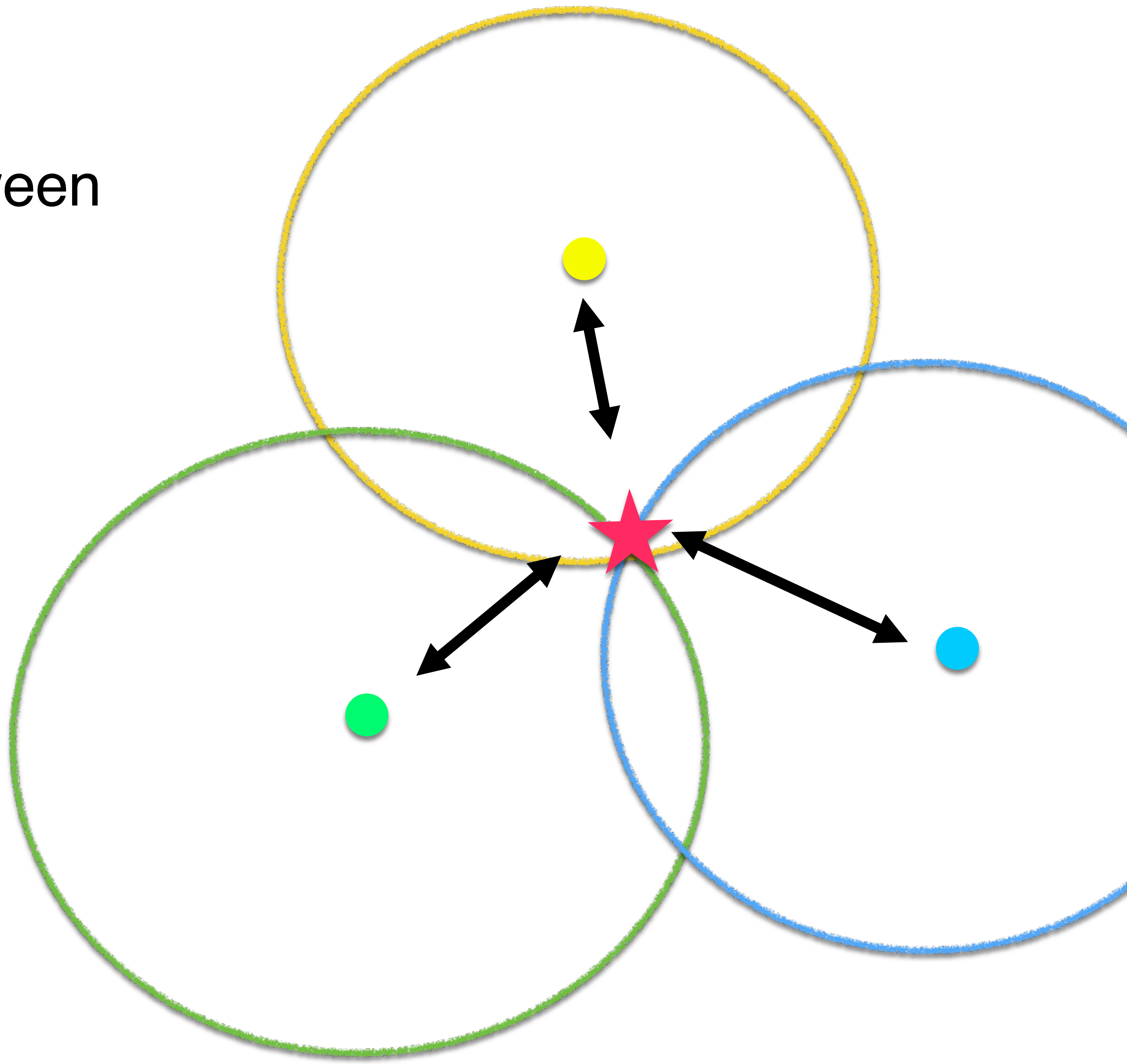
Attack Overview

- Focusing on fooling client side App located on users' mobile devices;
- Developing an automated system which can be easily scaled up:
  - Exploiting the localization service protocols to fake anchor points
  - Modifying Android framework to dump location readings in LBSNs



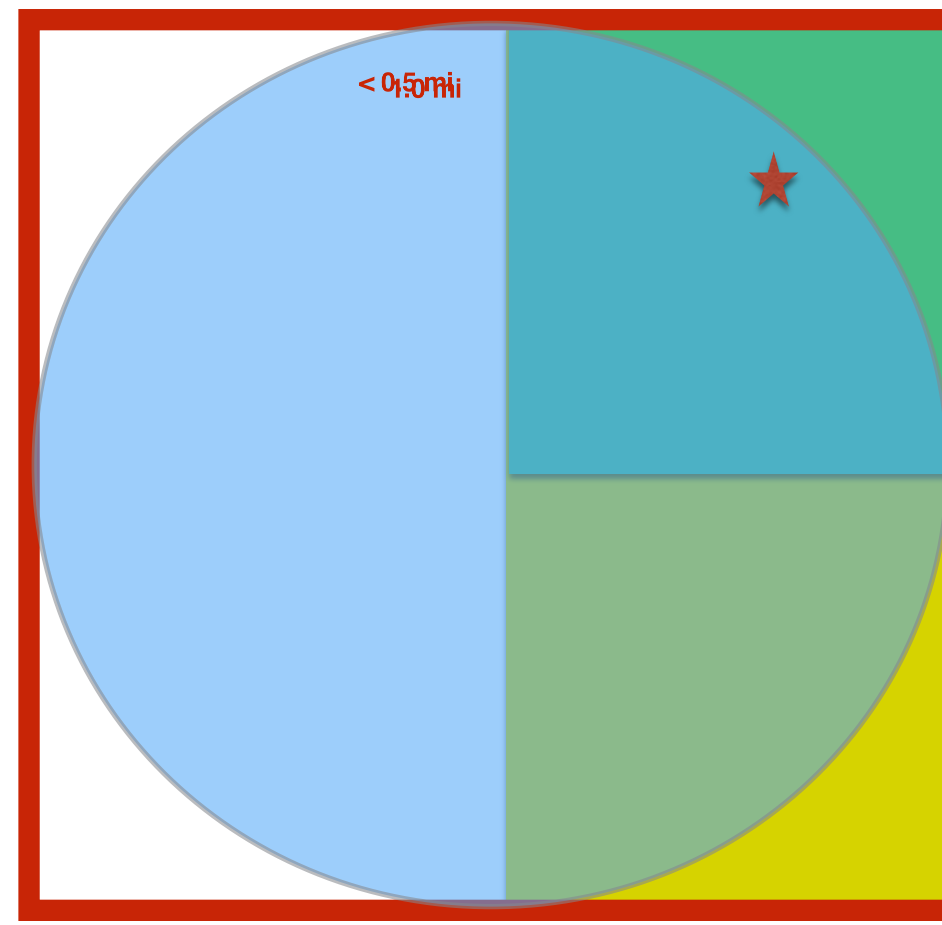
Attack Logic: Trilateration

- Momo shows relative distances between users:
  - Set up 3 anchor points
  - Trilateration the location of the victim
  - Iterate multiple rounds of trilateration to improve accuracy



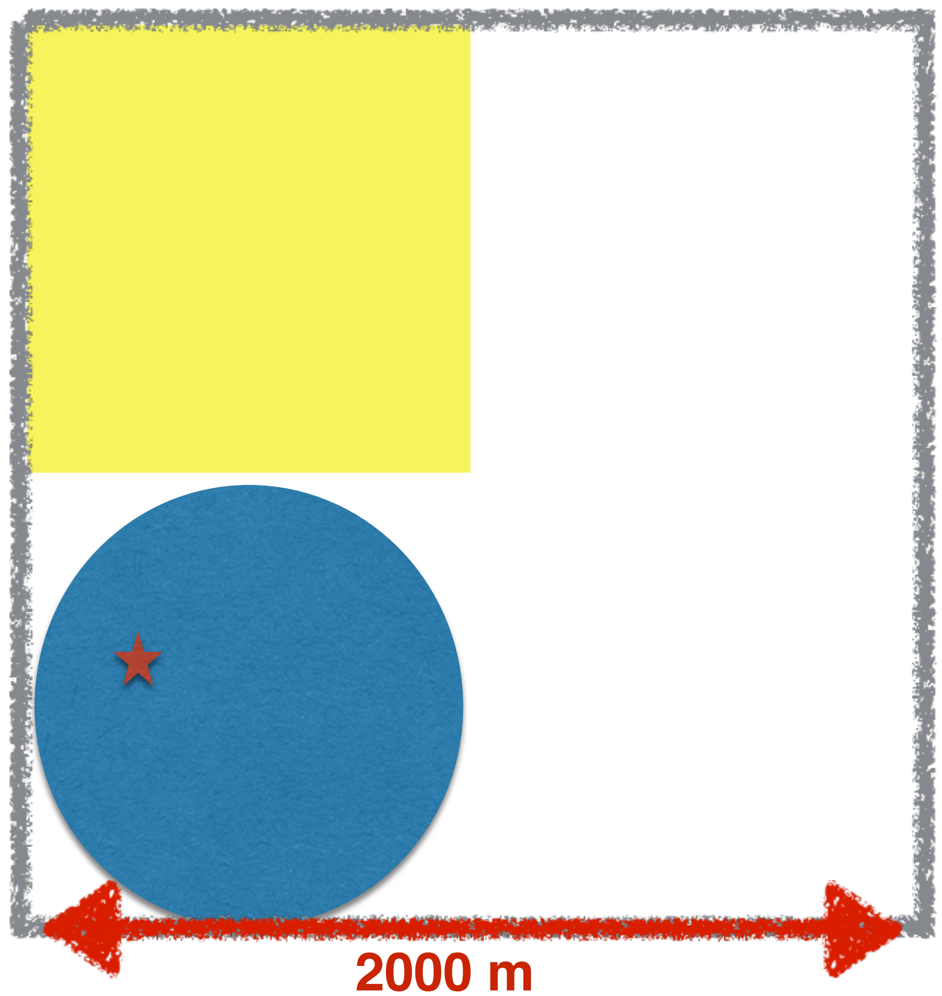
Attack Logic: Space partition

- Skout displays “< 0.5mi” instead of showing real distances when 2 users are within 0.5mi
- Partition the space based on this information to estimate the user’s location



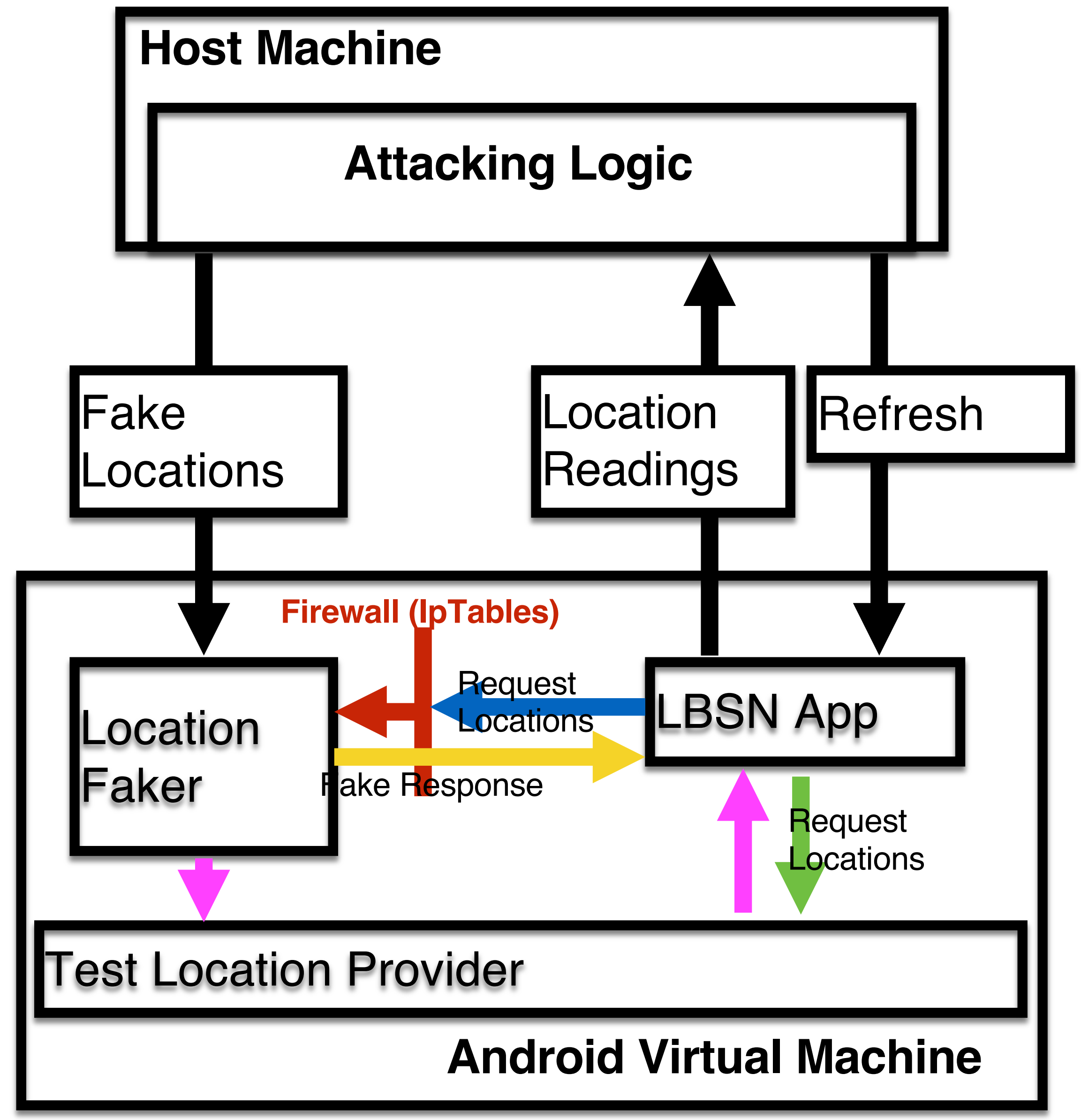
Attack Logic: Scan & Partition

- Wechat restricts visibility to 1km only
- Scan through the possible area with a 1km-step-size
- Then launch space partition to further improve precision



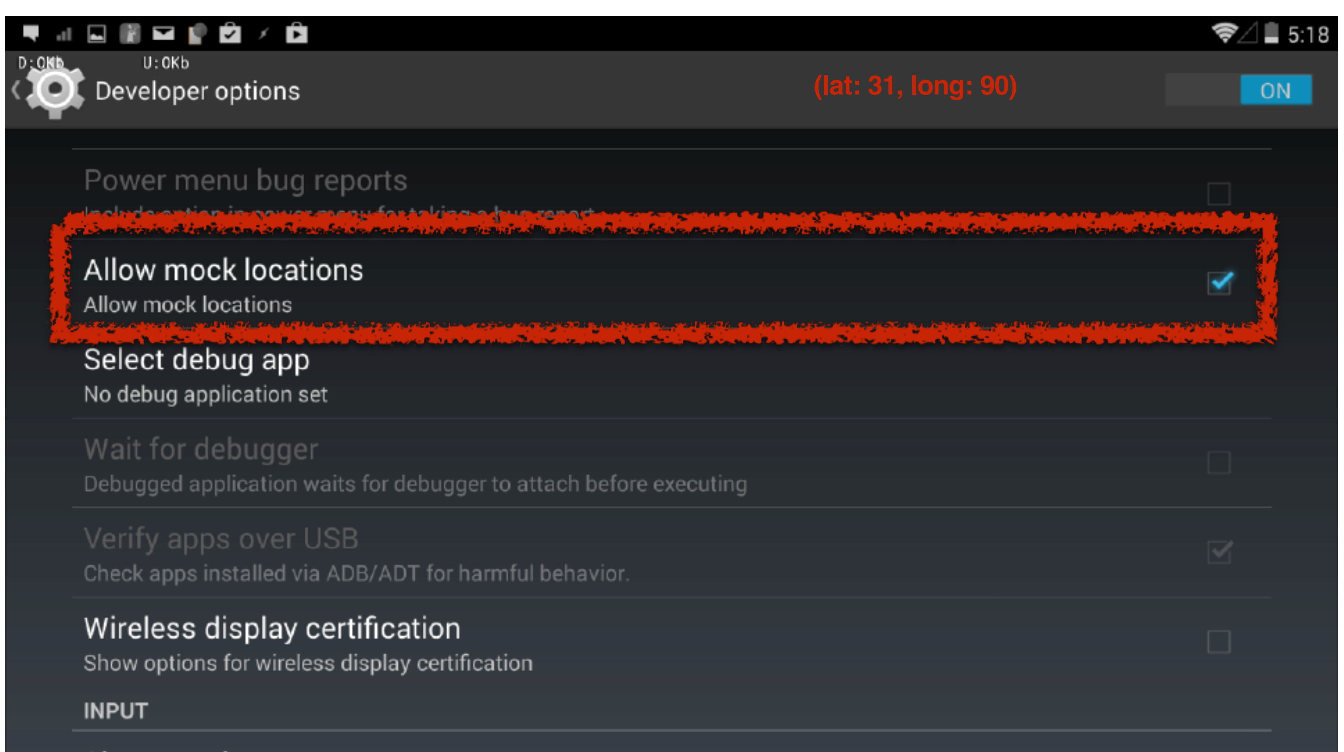
System Implementation

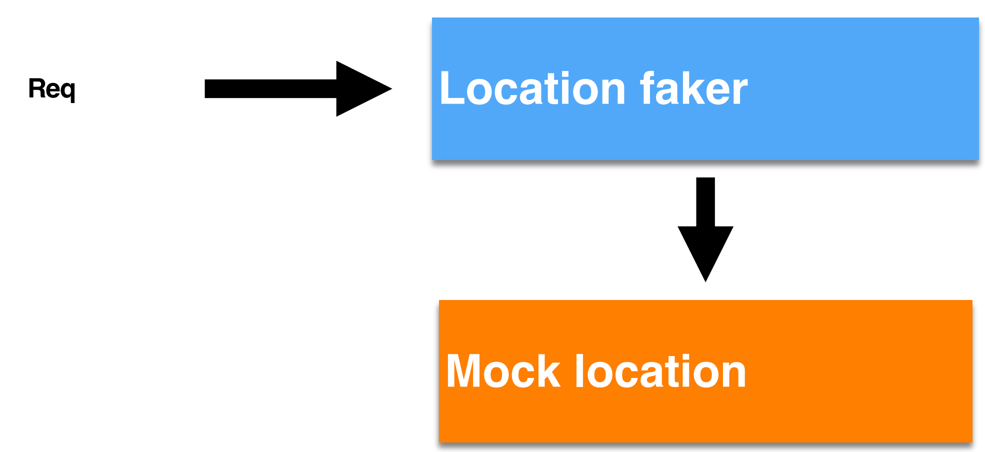
- The attacking logic runs inside the host machine:
  - Carry out localization calculations
  - Communicate with the location faker app in the VM to set fake locations
  - Trigger location updates in LBSN apps and retrieve location readings from Android's ADB logs
- The location faker app in the VM sets fake locations by:
  - Use Android's mock location provider
  - Act as a location server that answers the location requests from LBSN apps



Generating Testing Points: Mock Location

- Android system allows setting mock location via the test location provider for the debugging purpose
- Our Location Faker implements a test location provider
- The Location Faker can accept requests from the attacking logic unit to update locations

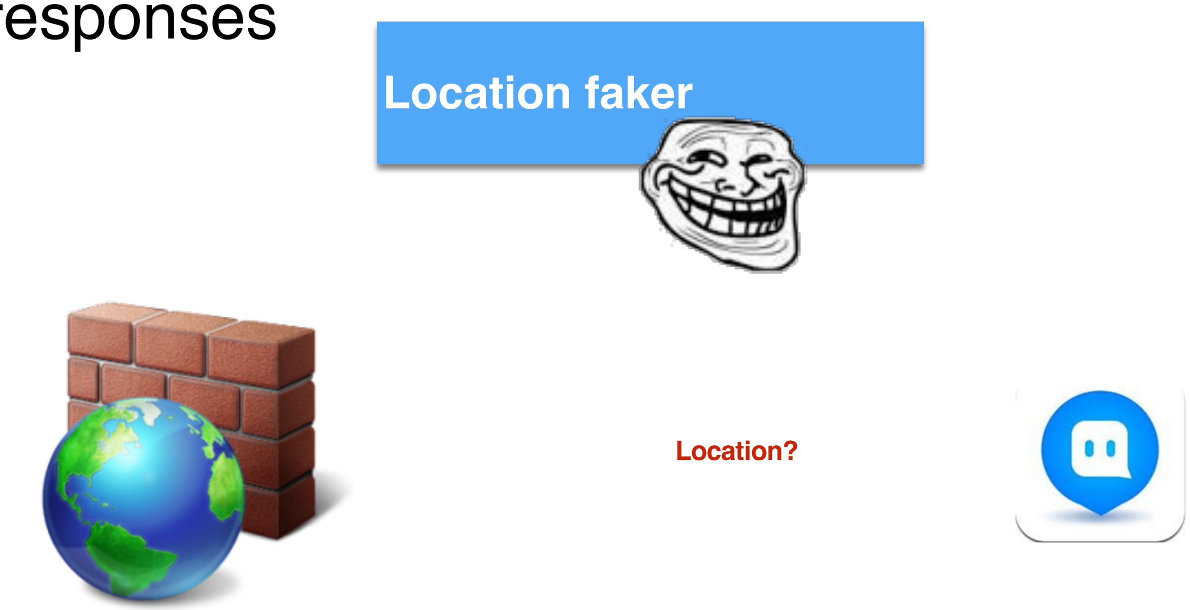




Generation of Test Points: Spoofing the Localization Protocol

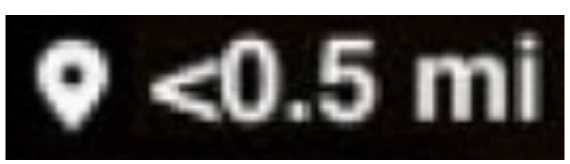
- Momo uses Baidu location API that does not allow taking in mock locations in VMs
- We intercept the network traffic and send fake response:
  - using the kernel firewall (IpTables) to intercept and redirect the location requests to Baidu location API servers
  - using our Location Faker to send fake responses

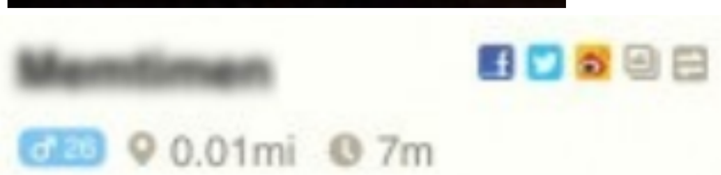
{"content":{"addr":{"detail":""},
"bldg":"","floor":"",
"point":{"y":"90","x":"31"},
"radius":""},
"result":{"error":,"time":""}}



Reading Locations from Apps in Android Framework

- All text related information is displayed in a widget called TextView provided by the Android framework
- The widget has an interface “TextView.SetText”, which is called by the apps to show texts
- We insert code in TextView.SetText function to dump text to the ADB log buffer





SetText(text)
SetText(text)
Log.d(text)
…

0.5 mi

Automating Mobile Location Updating Operations in LBSN Apps

- The location updating operation in LBSN apps consists of multiple tapping / dragging due to the screen size of the mobile devices
- We simulate these inputs to refresh the locations of the LBSN apps
- We mimic screen scrolling with multiple drags to deal with long user lists to read back all distance readings

Outline

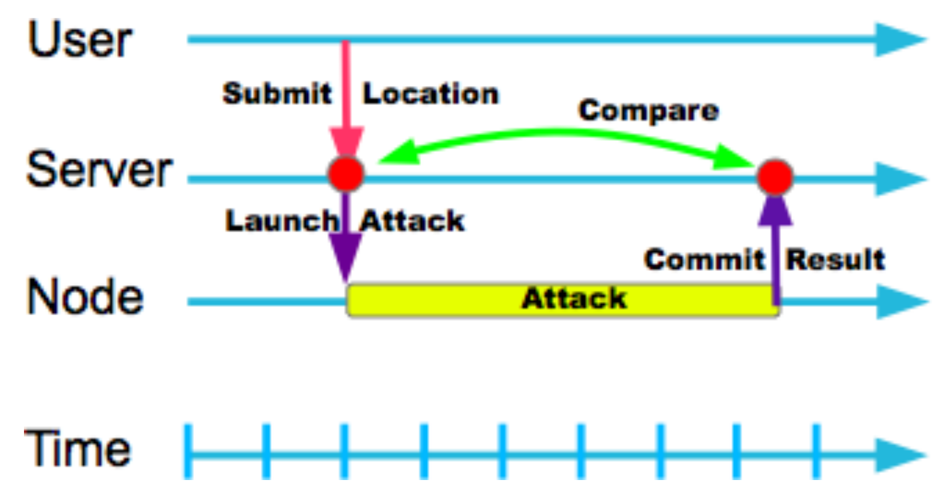
- Location Information Management in Mobile Social Networks
- Our Automated Attack Framework for User Location Discovery
- Our Real-world Attack Experiment and Results
- The Proposed Defense Mechanisms
- Concluding Remarks

Evaluation Overview

- We perform 3-week long evaluations with 30 volunteers from China, Japan, and U.S and focus on
  - Tracking accuracy:
    - Synchronous tracking accuracy measures the effectiveness of our localization strategy
    - Asynchronous tracking accuracy measures the effectiveness of the strategy in real world scenario
  - Localization efficiency and possible improvements
  - Effectiveness of long term tracking

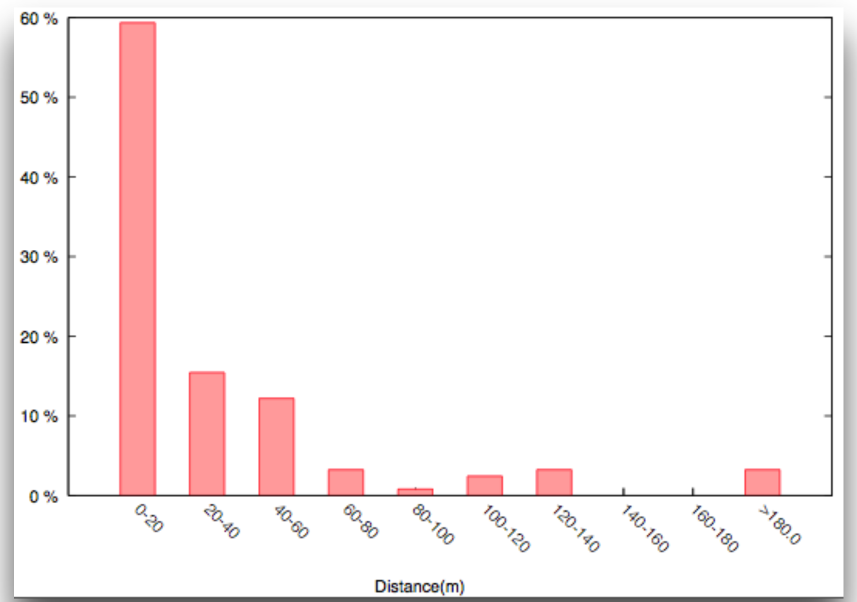
Accuracy: Synchronous Tracking

- Synchronous tracking measures the effectiveness of the localization strategy without the interferences from users' mobility
- Users refresh the location readings on LBSN apps and report their locations to the server
- Upon receiving a report, server launches an attack immediately
- The accuracy is measured as distance between the inferred location and the user's real location

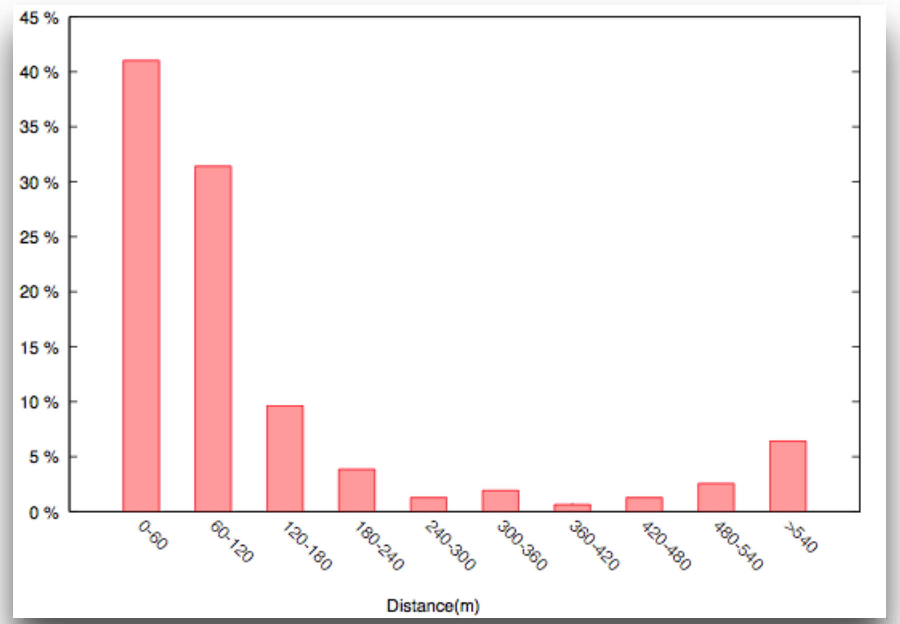


Synchronous Tracking Results

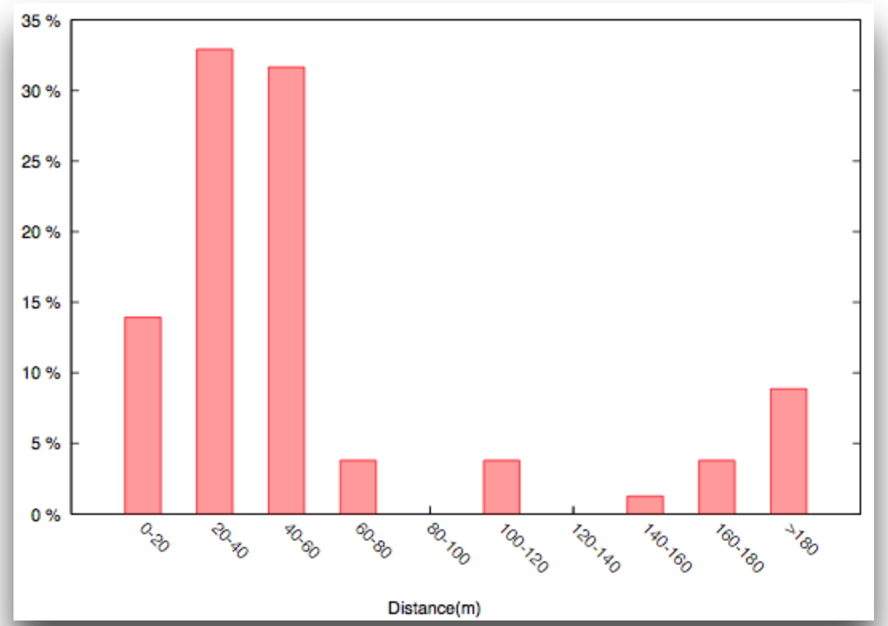
- Our tracking method achieves high accuracy for each of the LBSN applications
- It dramatically improves the accuracy compares to the location protection strategies



Momo



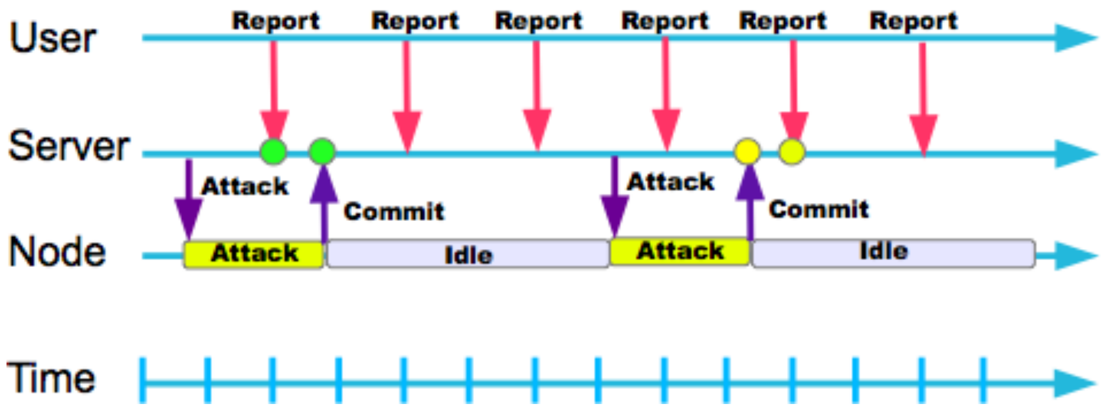
Skout



Wechat

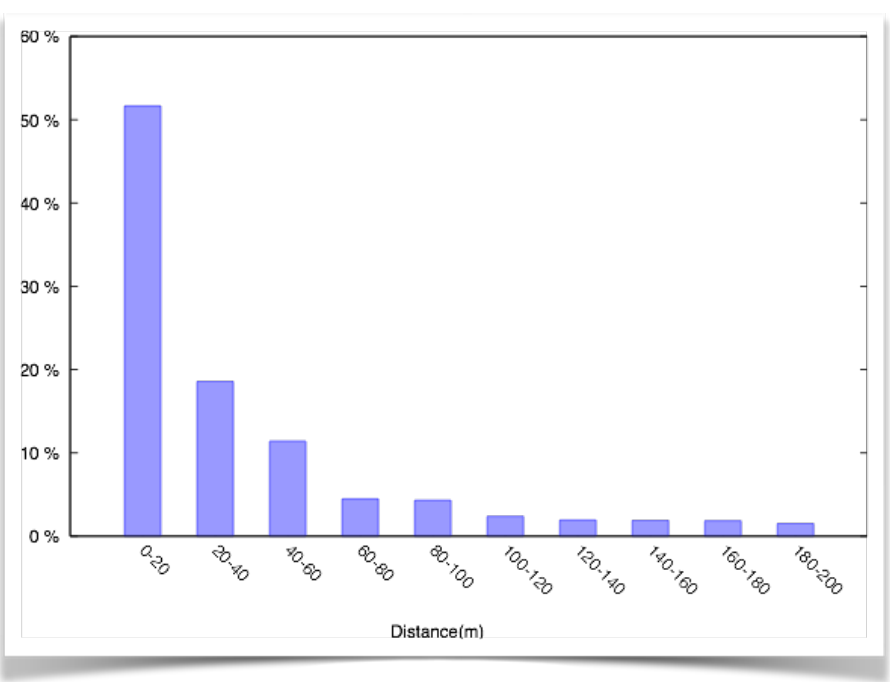
Accuracy: Asynchronous Tracking

- An user may move after refreshes location readings in LBSNs and before the attack
- The volunteers carry an app that automatically reports locations periodically
- Attacks are scheduled at a lower frequency
- We match the closest points in timeline and compare their distances to evaluate the tracking accuracy

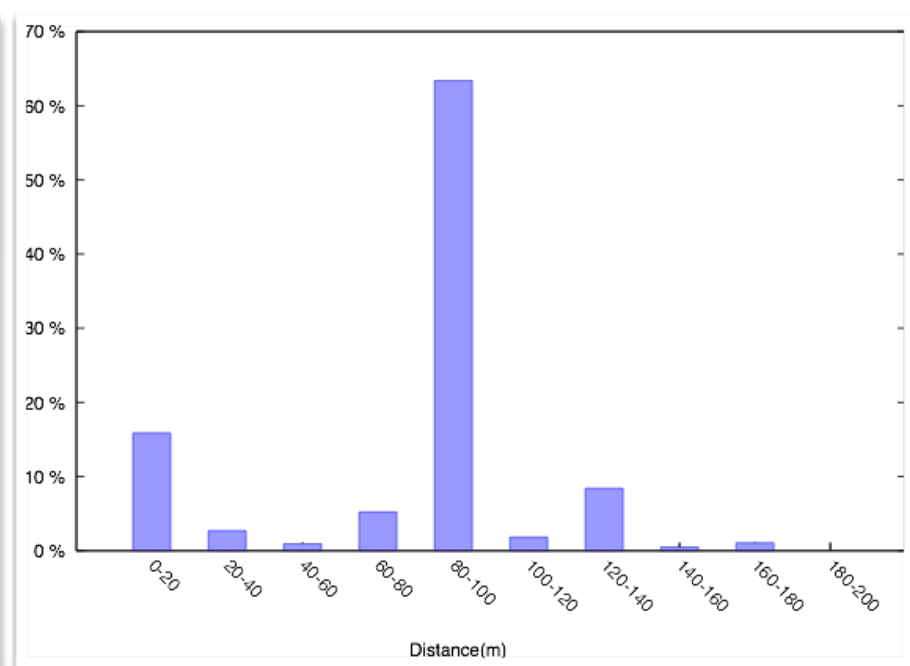


Asynchronous Tracking Results

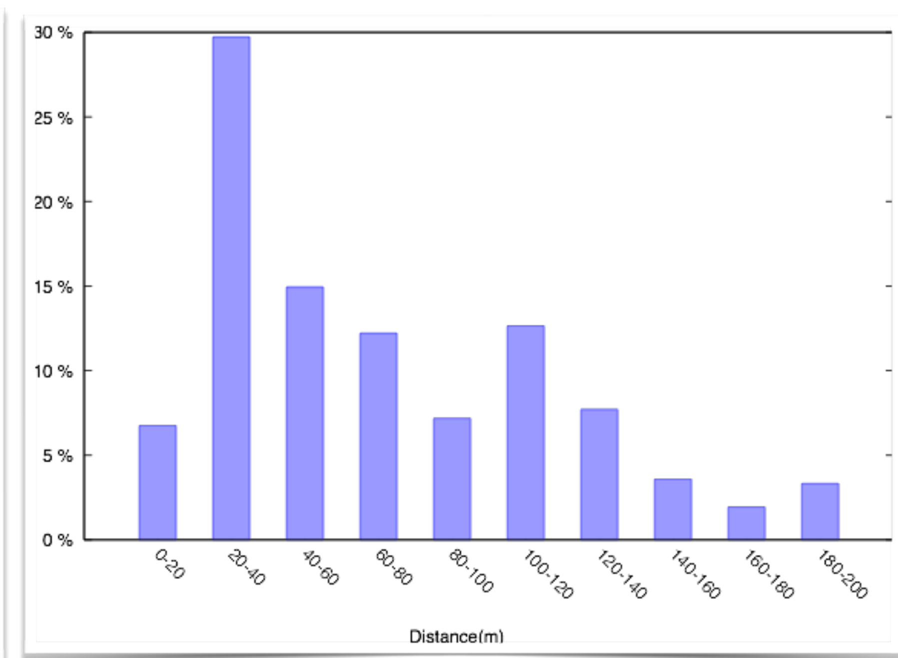
- Our tracking method is still significantly more accurate than the location protection constraints with user's mobility



Momo



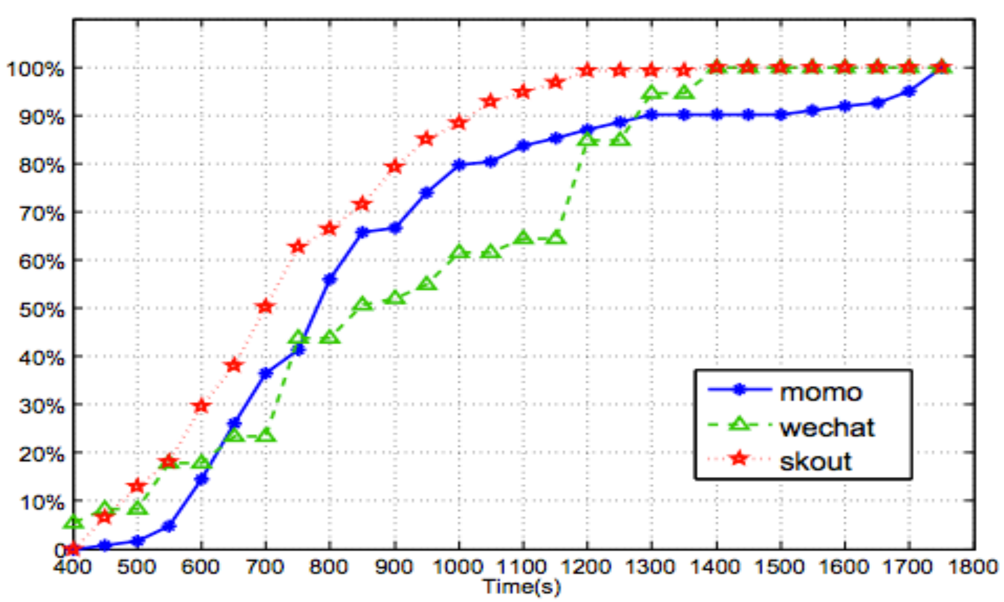
Skout



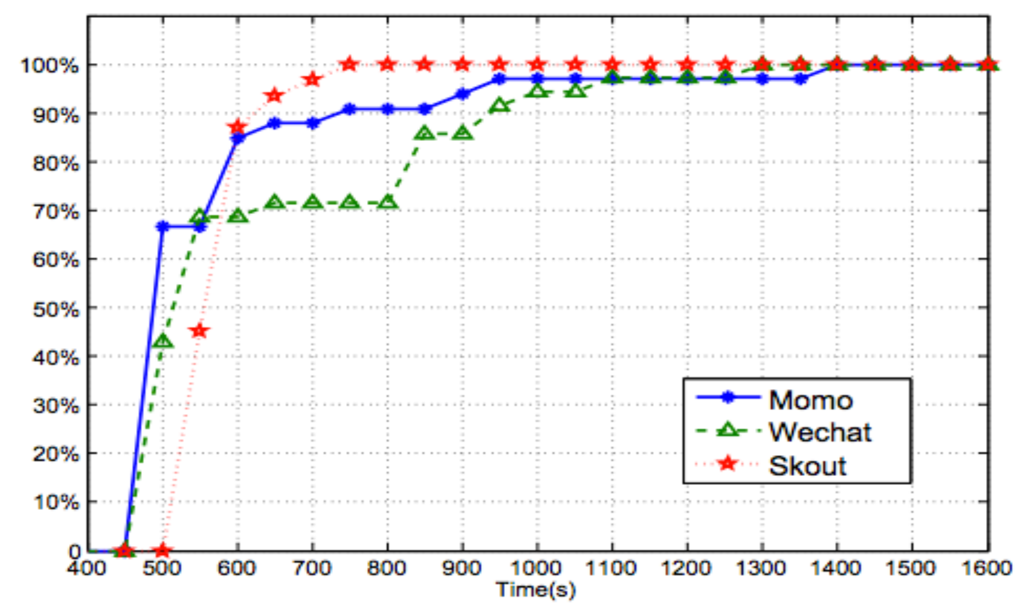
Wechat

Tracking Efficiency

- 80% of the attacks take < 900s to complete when anchor points are randomly chosen globally
- Most of the time is spent on waiting for network responses
- Efficiency can be dramatically improved with a little prior knowledge (e.g. the city in which the user is in and popularity distribution of the area)



CDF of Tracking Efficiency



CDF of Improved Tracking Efficiency

Effectiveness of Long-term Tracking

- Top-N locations refer to users N most frequently visited locations
- Existing works show that Top-N locations are closely related to a user's identity [1]
- We evaluate how many Top-N locations are revealed in our 3-week tracking

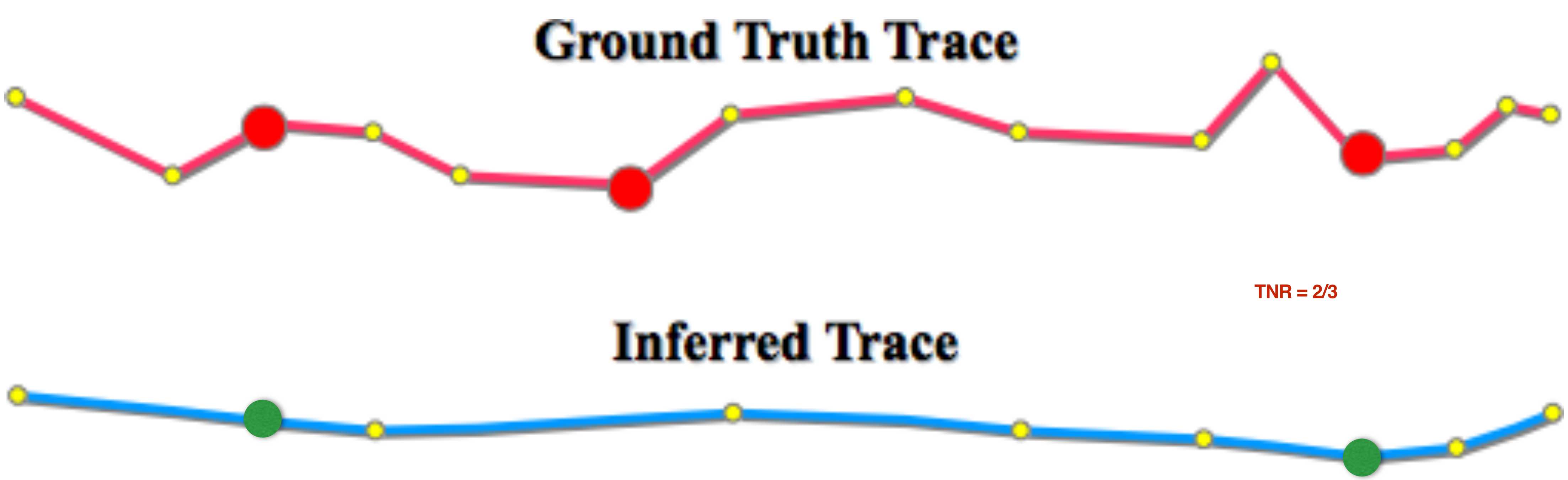
[1] Unique in the Crowd: The privacy bounds of human mobility. de Montjoye et. al. Nature. 2013

Top-N Location Coverage

- Top-N location coverage rate is defined as:

$$TNR = \frac{|Top_N(\mathbb{G}) \cap Top_N(\mathbb{I})|}{N},$$

G: Ground truth traces
I: Inferred traces



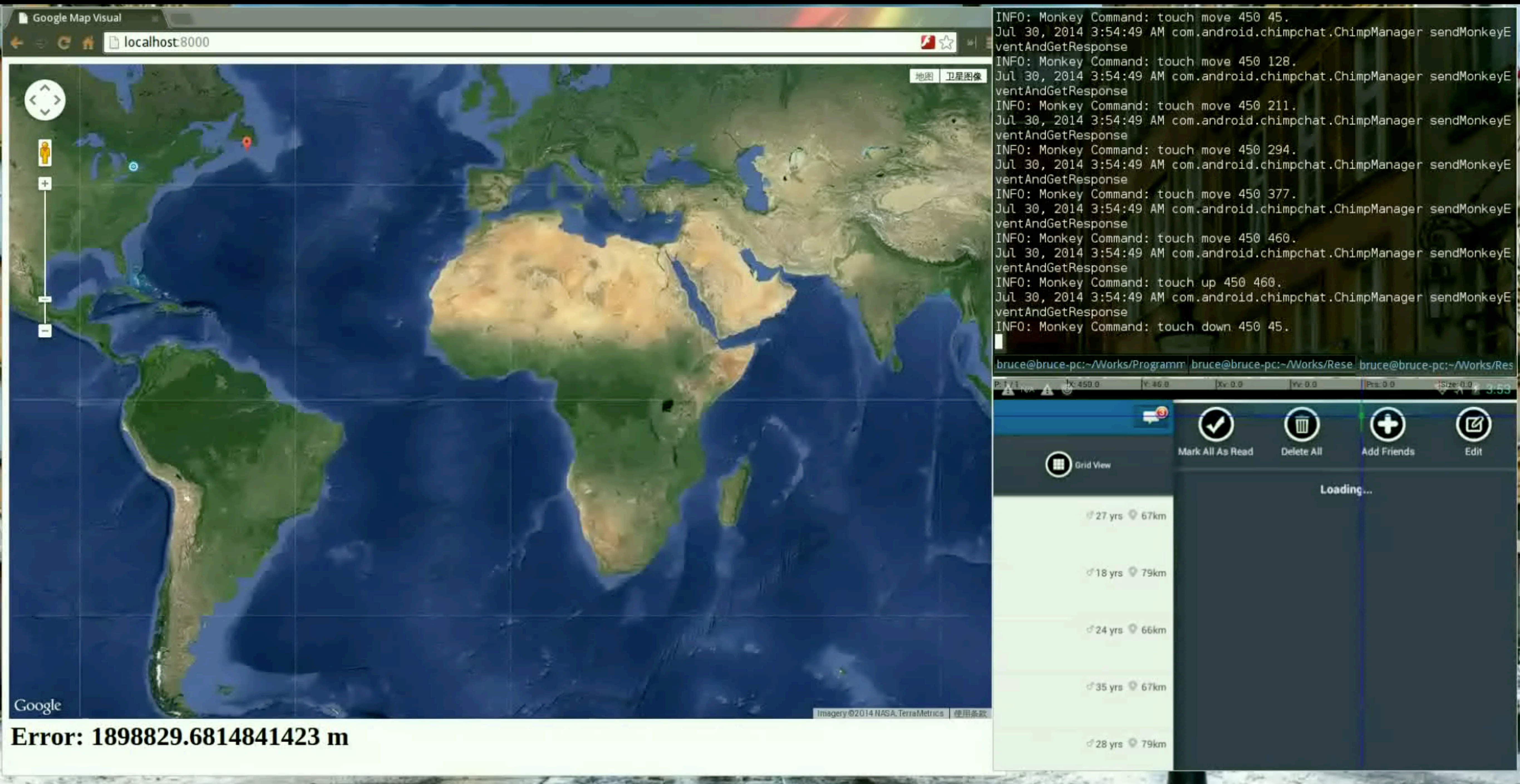
Top-N location coverage

- Top-N location coverage rate grows in 3-week experiments
- For all 3 apps, we achieve high Top 1 location coverage rate
- Our top locations are much finer-grained than existing works [1][2]

Top locations	1 week			2 weeks			3 weeks		
	Momo	Skout	Wechat	Momo	Skout	Wechat	Momo	Skout	Wechat
1	92.3%	20.0%	50.0%	100.0%	60.0%	57.1%	100.0%	60.0%	71.4%
2	46.1%	0.0%	21.4%	46.1%	40.0%	21.4%	69.2%	40.0%	21.4%
3	30.7%	20.0%	21.4%	46.1%	60.0%	28.5%	38.4%	80.0%	28.5%
4	23.0%	20.0%	35.7%	30.7%	40.0%	35.7%	38.4%	40.0%	35.7%
5	23.0%	0.0%	21.4%	15.3%	40.0%	21.4%	15.3%	40.0%	14.2%

[1] Unique in the Crowd: The privacy bounds of human mobility. Y. Montjoye et al. Nature. 2013
[2] Anonymization of Location Data Does Not Work: A Large-Scale Measurement Study. H. Zang et al. MobiCom'13.

Attack Demo



Google Map Visual
localhost:8000
地图
卫星图像
Google
Imagery ©2014 NASA, TerraMetrics
使用条款
Error: 1898829.6814841423 m
INFO: Monkey Command: touch move 450 45.
Jul 30, 2014 3:54:49 AM com.android.chimpchat.ChimpManager sendMonkeyEventAndGetResponse
INFO: Monkey Command: touch move 450 128.
Jul 30, 2014 3:54:49 AM com.android.chimpchat.ChimpManager sendMonkeyEventAndGetResponse
INFO: Monkey Command: touch move 450 211.
Jul 30, 2014 3:54:49 AM com.android.chimpchat.ChimpManager sendMonkeyEventAndGetResponse
INFO: Monkey Command: touch move 450 294.
Jul 30, 2014 3:54:49 AM com.android.chimpchat.ChimpManager sendMonkeyEventAndGetResponse
INFO: Monkey Command: touch move 450 377.
Jul 30, 2014 3:54:49 AM com.android.chimpchat.ChimpManager sendMonkeyEventAndGetResponse
INFO: Monkey Command: touch move 450 460.
Jul 30, 2014 3:54:49 AM com.android.chimpchat.ChimpManager sendMonkeyEventAndGetResponse
INFO: Monkey Command: touch up 450 460.
Jul 30, 2014 3:54:49 AM com.android.chimpchat.ChimpManager sendMonkeyEventAndGetResponse
INFO: Monkey Command: touch down 450 45.
bruce@bruce-pc:~/Works/Programm
bruce@bruce-pc:~/Works/Rese
bruce@bruce-pc:~/Works/Res
Mark All As Read
Delete All
Add Friends
Edit
Grid View
Loading...
27 yrs
67km
18 yrs
79km
24 yrs
66km
35 yrs
67km
28 yrs
79km

Outline

- Location Information Management in Mobile Social Networks
- Our Automated Attack Framework for User Location Discovery
- Our Real-world Attack Experiment and Results
- The Proposed Defense Mechanisms
- Concluding Remarks

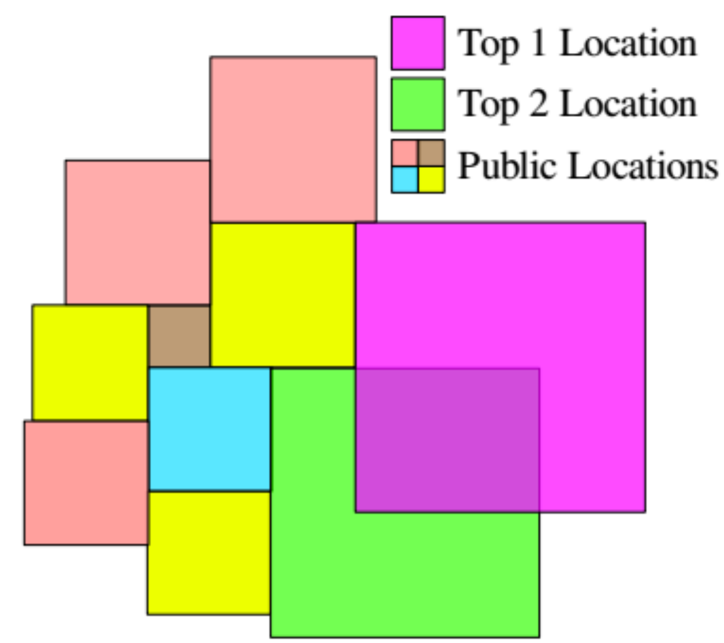
Defense Mechanism Overview

- One possible defense mechanism is to use location obfuscation
- We outlined a user-centric location obfuscation mechanisms to achieve a good balance between utility and effectiveness:
  - More obfuscation when users are at their Top-N locations
  - Less obfuscation when users are at public places
- We implement this technique as an Android location service

User-Centric Location Obfuscation

- We record users' location profiles and identity Top-N locations

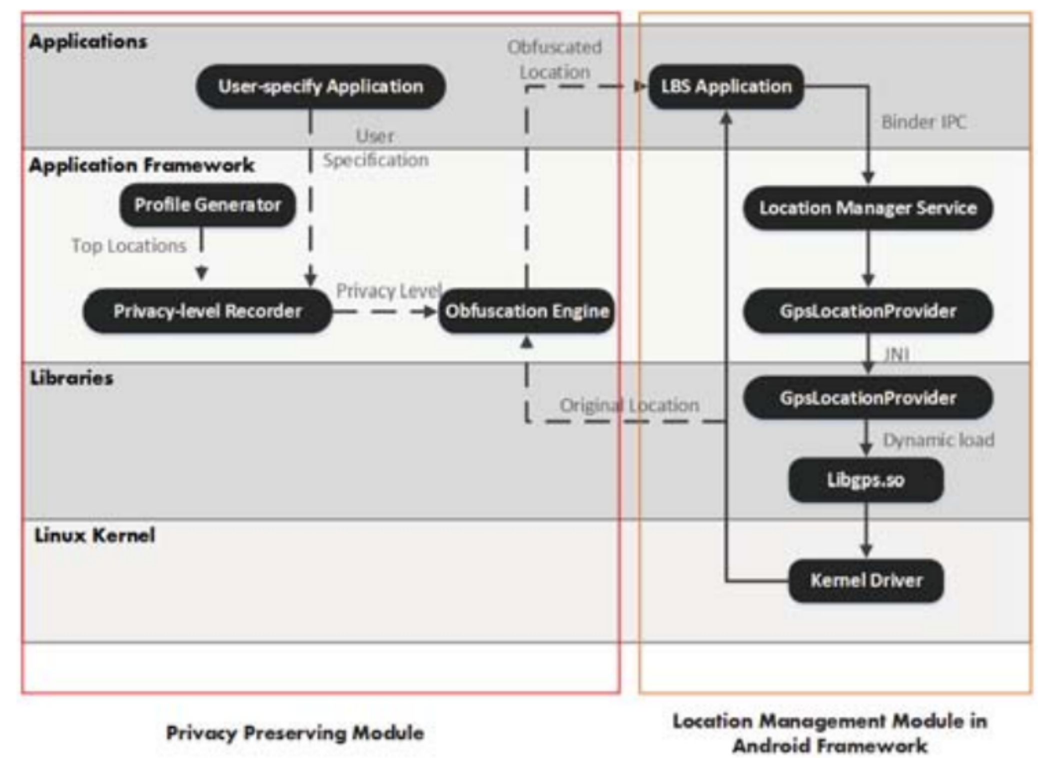
- Users selectively apply:
  - more obfuscation to Top-N locations
  - less obfuscation to public places



Implementation as Android Location Service

- The profile generator collects and identifies Top-N locations

- The obfuscation middleware intercepts location requests from the applications and replies with obfuscated locations



Concluding Remarks

- We have developed automated attacks for the first time against popular LBSNs with hundreds of millions of users
- Proximity-based friend discovery poses serious threats to users' location privacy
- Automated tracking attacks without hacking into LBSN services can be carried out without much technical difficulty and resource.

Concluding Remarks

- It is very important to protect users' location privacy in today's world
- We believe that people should be able to take the control of their own personal location data
- Open-source "personal location obfuscater" controlled only by the user him/herself is desired:
  - Continuously learn his/her own location profile
  - Perform adaptive location obfuscation on-demand to all mobile apps that request user location info based on
    - the nature of the app and
    - his/her own location profile

Q & A