AFRL-RI-RS-TR-2022-158

INCREASING PRIVACY FOR ANDROID (IPA)

TWO SIX TECHNOLOGIES

NOVEMBER 2022

FINAL TECHNICAL REPORT

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

STINFO COPY

AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE

■ **AIR FORCE MATERIEL COMMAND** ■ **UNITED STATES AIR FORCE** ■ **ROME, NY 13441**

NOTICE AND SIGNATURE PAGE

Using Government drawings, specifications, or other data included in this document for any purpose other than Government procurement does not in any way obligate the U.S. Government. The fact that the Government formulated or supplied the drawings, specifications, or other data does not license the holder or any other person or corporation; or convey any rights or permission to manufacture, use, or sell any patented invention that may relate to them.

This report was cleared for public release by the 88th ABW, Wright-Patterson AFB Public Affairs Office and is available to the general public, including foreign nationals. Copies may be obtained from the Defense Technical Information Center (DTIC) (http://www.dtic.mil).

AFRL-RI-RS-TR-2022-158 HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION IN ACCORDANCE WITH ASSIGNED DISTRIBUTION STATEMENT.

FOR THE CHIEF ENGINEER:

/ S /
CARL R. THOMAS
Work Unit Manager

/ S /
GREGORY J. HADYNSKI
Assistant Technical Advisor
Computing & Communications Division
Information Directorate

This report is published in the interest of scientific and technical information exchange, and its publication does not constitute the Government's approval or disapproval of its ideas or findings.

REPORT DOCUMENTATION PAGE

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED	
		START DATE	END DATE
NOVEMBER 2022	FINAL TECHNICAL REPORT	SEPTEMBER 2018	JUNE 2022

4. TITLE AND SUBTITLE

INCREASING PRIVACY FOR ANDROID (IPA)

5a. CONTRACT NUMBER	5b. GRANT NUMBER	5c. PROGRAM ELEMENT NUMBER
FA8750-18-C-0170	N/A	62303E

5d. PROJECT NUMBER	5e. TASK NUMBER	5f. WORK UNIT NUMBER
		R2MF

6. AUTHOR(S)

Daniel Hallenbeck

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Two Six Technologies
901 N Stuart St, Suite 1000
Arlington, VA 22203

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Air Force Research Laboratory/RITA
525 Brooks Road
Rome NY 13441-450

DARPA
675 North Randolph St
Arlington, VA 22203-2114

10. SPONSOR/MONITOR'S ACRONYM(S)

AFRL/RI

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

AFRL-RI-RS-TR-2022-158

12. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for Public Release; Distribution Unlimited. PA#: AFRL-2022-5681
Date Cleared: 23 November 2022

13. SUPPLEMENTARY NOTES

14. ABSTRACT

Throughout the world smartphones are being carried and used by all types and ages of people. These always on devices are equipped with numerous hardware chips and sensors such as GPS, Wifi, Bluetooth, and cellular modems, all of which can be utilized to track and report location. The ability of mobile applications to persistently monitor, track, and record the devices location poses a serious privacy risk, not only to normal cell phone users, but also to US government personnel. Increasing Privacy for Android (IPA) is a custom Android OS that aims to address this privacy risk posed by modern smartphones.

15. SUBJECT TERMS

Privacy, Android, Location, Spoofing

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES
a. REPORT	b. ABSTRACT	C. THIS PAGE		
U	U	U	SAR	

19a. NAME OF RESPONSIBLE PERSON	19b. PHONE NUMBER *(Include area code)*
CARL R. THOMAS	N/A

PREVIOUS EDITION IS OBSOLETE.

STANDARD FORM 298 (REV. 5/2020)
Prescribed by ANSI Std. Z39.18

Table of Contents

List of Figures

List of Tables

1 Summary

Throughout the world smartphones are being carried and used by all types and ages of people. These always on devices are equipped with numerous hardware chips and sensors such as GPS, Wifi, Bluetooth, and cellular modems, all of which can be utilized to track and report location. The ability of mobile applications to persistently monitor, track, and record the devices location poses a serious privacy risk, not only to normal cell phone users, but also to US government personnel.

Increasing Privacy for Android (IPA) is a custom Android OS that aims to address this privacy risk posed by modern smartphones. When configured, IPA can comprehensively report a location to requesting applications that is different than the devices actual location. It does this by providing a specific GPS coordinate, as well as reporting Wifi and cell towers scans that would be observed at the desired location. This comprehensive spoofing of the devices location data is accepted by the applications on the phone, as well as backend services, such as Google Location History. As a result, the user's privacy is preserved through preventing the disclosure of the users actual location when that is desired.

2 Introduction

The increasing pervasiveness of smartphones across the globe, among all populations and age groups, has also been accompanied by a decrease in personal privacy. A smartphone, with its numerous sensors, constantly carried around by its owner, has essentially become a high-powered tracker. A study by Exodus Privacy and the Yale Privacy Lab showed that 75% of Android apps track users with 3rd party tools [1]. Additionally, aside from the privacy issues that smartphones present, there are real security risks to US service personnel as highlighted by the recent disclosure of US bases through the Strava fitness app and service [2].

Many of the existing security and privacy solutions for mobile devices do not adequately reduce the privacy risks associated with apps collecting a user's private information that may be used to track an individual. For example, Android OS replacements like Copperhead OS attempt to prevent the exploitation of the device to prevent privacy leakages but do nothing to prevent apps that request private information from obtaining it.

One of the more sensitive, and yet harder to disguise, types of information generated by mobile phones is location data. Android phones in use today typically have at least three sources of location information: a GPS, Network, and Fused Location Provider. The GPS Location Provider gathers, and reports information solely obtained from the GPS sensor on the phone (implemented in the Android OS). The Network Location Provider utilizes both cellular network and Wi-Fi data information to report a phone's location (implemented by carrier or manufacturer). The Fused Location Provider, typically Google Play Services, gathers data from multiple sources (GPS, Wi-Fi, Bluetooth, cellular network info, etc.) and combines them to provide a more accurate location when one or more data sources may be disrupted.

The goal for Increasing Privacy for Android IPA is to provide comprehensive, consistent, and realistic location information to requesting applications without their knowledge, thereby protecting the privacy of location

information for the user of the device. As such, IPA is developed as a custom Android OS to ensure the appropriate level of access to system services allowing it to shim in the desired information prior to it being sent to the requesting applications. However, IPA does not try to mask phone activity on the physical cellular infrastructure, nor does it prevent IP based geo-location by backend services.

3 Methods, Assumptions, and Procedures

To begin to evaluate and test the feasibility of providing alternative location data to applications, Two Six had to choose a platform to build on. The initial platform chosen was the Google Pixel 3a XL running the Android Open Source Project (AOSP) version 9 for the Operating System. Eventually, to demonstrate the portability of the approach, the core changes and modifications were ported to AOSP version 10, and Lineage 17.1 versions of the OS, running on the Google Pixel 4a, One Plus Nord, and the Samsung S10e in addition to the original target of the Google Pixel 3a XL. Table 3-1 depicts the variations supported by IPA for hardware and OS versions.

Table 1- IPA Phone and OS variations. Note, "y" indicates some cellular functionality is missing like VoLTE or RCS

Platform	Pixel 3aXL	Pixel 4a	One Plus Nord	Samsung S10e
AOSP 9 (Pie)	X			y
AOSP 10	X	X	X	y
Lineage 17.1	X	X	X	

Once the development platform was established, it was necessary to identify the appropriate hook points within the base OS (AOSP or Lineage) to insert code that enabled IPA to report a desired location or location based information. It should be noted, that as Android versions advance, or if porting to different variants of Android, the hook points may change or there may be a need to add more hook points to protect location sensitive information. This evaluation and survey of the source code is something that should be done for each new version that IPA is ported to. For AOSP version 9, Two Six identified the below classes as required hook points for GPS, Wifi scans, cell scans, and Bluetooth scans:

- LocationManagerService (GPS and other location provider reporting)
- GeofenceProxy (Hardware based GPS geo-fence trigger)
- WifiServiceImpl (Wifi scans)
- WifiScanningServiceImpl (Wifi scans)
- PhoneInterfaceManager (Cell scans)
- GattService (Bluetooth scans)
- RemoteDevices (Bluetooth scans)

From there, Two Six developed the IpaLocationService to act as the control center of location data for the IPA phone. This service reads and loads IPA configurations as well as maintain the state of IPA by monitoring triggers to start and stop the IPA location reporting. It also maintains a list of applications that are allowed to access the phones actual location. Further, it allows for the individual blocking of GPS, Wifi Scans, or cellular scans. For generating location data, IpaLocationService either reports a desired location as specified by the configuration file, or based on open source data sets such as Open Street Maps [3], Wigle Wifi data [4], or Mozilla cellular data [5].

With the IPA platform and software in place, the next step is to enable testing the functionality and efficacy of IPA to the extent possible with the tools at our disposal. The first step to enable this testing was to install and configure applications and services within the phone that could later be utilized for verification. To that end, Two Six signed up for several Google accounts, and enabled location history, high accuracy location services, and wireless scanning on the phones. This allowed Two Six to monitor the perceived location of the IPA devices associated with the account. Two Six also utilized running/workout apps, like Strava, to record IPA reported tracks, as well as several mapping applications to observe real time location as reported by IPA. Finally, IPA was configured to write the real and projected location to a log file for later analysis and corroboration with external services.

Next, Two Six developed IPA configuration and testing tools. The IPA configuration tool was developed as an Angular based web app that allowed for the easy creation and deployment of IPA configuration files to phones. Further, it enabled a user to create test tracks that could be "played" on the phone. These test tracks enabled a user to test the IPA software by playing an artificial track on the phone that triggered the IPA software without actually needing to be at a particular location.

4 Results and Discussion

Two Six approached testing the IPA software in incrementally complex scenarios. Initial tests began by testing whether IPA could report static locations, then dynamic location updates, and ultimately more realistic dynamic location updates as specified by specific reporting algorithms. At each stage Two Six would test using commercial apps from the Google Play store, locally developed test apps, and to the extent possible review data sent to back end services. The subsequent sections describe the results and nuances discovered while testing IPA.

4.1 Static Location Reporting

Initial tests of IPA focused on testing static location reporting functionality to demonstrate that the IPA OS modifications and hooks worked appropriately. Two Six was able to easily change the reported location merely

by changing reported GPS latitude and longitude, and leaving bearing, speed, and accuracy alone. Further, we discovered that reported GPS location seemed to override location information derived by other sources. Specifically, Wifi and cell scan info indicating locations other than the IPA reported GPS location were not considered or displayed in the applications tested. Through further testing where we systematically disabled GPS, Wifi, or cell scan reporting, we observed that Wifi scans trump cell scans as well. As such, the empirically observed preference for location information by observed applications is first GPS, then Wifi scans, then cell tower scans.

It should be noted, that while doing the static location tests, some applications did not immediately report the updated location. For example, some weather applications seemed to only check/update the weather based on location approximately every 15-20 minutes. Additionally, Google Maps would “skip” or stutter to the newly reported location, rather than instantly jumping. Two Six hypothesizes that this is some form of location smoothing to help display smooth movement while driving.

4.2 Dynamic Location Reporting

After Two Six saw successful results to static location reporting, we moved on to dynamic location reporting, or showing movement within a target area. The first approach here was to mimic directionality and speed within a target location. This was done by computing a simple offset from the current location to the target location. With each location update that came in from the real sensors, IPA would apply the offset and report the updated location to apps. This approach also proved successful, and applications did report similar movement within the target area, as shown in Figure 1 below. However, cursory analysis of the reported route would likely raise red flags as the route shows the phone crossing a river and walking through buildings.

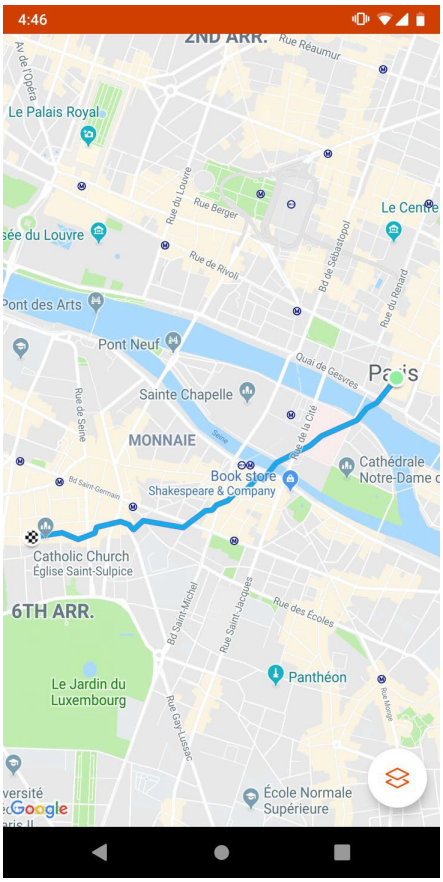

Figure 1- IPA reported location mimicking speed and direction

To address the issue of suspicious movement in the target area, Two Six began to leverage Open Street Maps data. Two Six modified the algorithm to only project a location on a road or footpath in the target location. This allowed the projected path to mimic the actual speed and keep a general direction. For example, if the actual path moved in a Northwest direction, the projected path would try to do the same, but ensure all travel is done on the Open Street Map provided road network. While this also worked as intended, there were still some strange artifacts. The algorithm would have to decide what path to take at intersections in the target area, however, later direction changes could cause the reported location to backtrack down a road to get back to the intersection and take another route. Additionally, Two Six found that some locations had missing or incomplete road networks that would cause our algorithm to get "stuck" and not be able to leave an area.

During this testing, Two Six would often project their location to be a good distance away from their actual location, often picking New York City, London, or Paris. While this effectively demonstrated the location reporting on a target road network, it also helped us verify that backend services with which applications talk, were also receiving the reported location. One example of this is Google's Location history timeline. Upon reviewing the timeline, we saw the reported locations within our test user's Location History data and timeline. Additionally, on a monthly basis, Google would send emails indicating the places we traveled and approximate total distance that our test user has travelled in all time. In July 2019, after a few months of testing, Google reported that our test account had traveled ~92,000 miles, or we still needed about 146,535 miles of 238,855 to travel the distance to the moon (see Figure 2). This large amount of travel is likely a byproduct of us jumping large distances to NYC, London, and Paris to conduct the testing and is a further indication that the data we reported to applications is in fact being received by backend services.

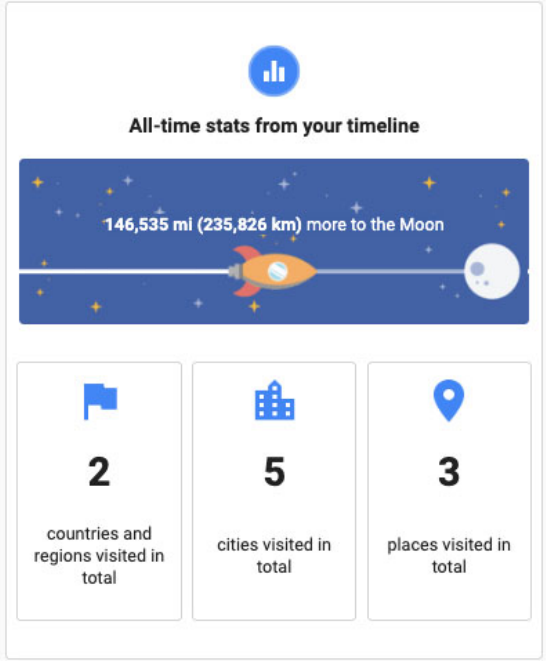

Figure 2- Email indicating distance travelled to the moon

4.3 Advanced Location Algorithms

After successfully demonstrating the IPA location reporting functionality, Two Six worked closely with potential end users to craft potential use cases and CONOPs for the IPA technology. By the end of the research there were four distinct modes that IPA supported; Pause/Unpause, Geofence, Dither, and Orchestration modes. All of the modes continued to report the specified location as expected, but each mode was intended to be used in different situations to better conceal specific, real movements. These modes are discussed below.

4.3.1 Pause/Unpause Mode

Pause/Unpause mode simply allows a user to "pause" at a particular location where the phone will only report that static location with some added noise. The IPA device will continue reporting until the user "unpauses" the device. When the IPA device is unpaused, it will do one of two distinct actions: either a) report the phone's actual location; or b) attempt to catch-up to the phone's current location using the road network. The first case occurs when the user unpauses the phone within 15m of the paused location, causing the phone to immediately jump to the current actual location and resume normal location reporting activities.

The second, the phone will leverage Open Street Maps to try to catch-up to the current location by traversing the road network. This catch-up can either happen at walking speed or driving speed. In either case, the actual
speed to catch-up will be 1.5x faster than the defined walking or driving speed. This was deemed the only way to possibly catch-up to a phone that is actually moving. Further, if the phone leaves the area for which it has road network data, IPA will immediately report the user's current location when the algorithm has reached the edge of the road network, making the phone appear to teleport.

4.3.2 Geofence Mode

Geofence mode is intended to prevent the phone from being reported within a particular area. Under geofence mode, when the phone enters the restricted area, it will use the road network to travel to and dwell at the closest defined Point of Interest (POI). While dwelling at the POI, IPA will introduce some noise/jitter to give the appearance of poor GPS/location accuracy. Similar to Pause/Unpause, when the phone leaves the geofence, IPA's algorithm will attempt to catch-up to the phone with the same 1.5x faster travel speed. It is worth noting, that the 1.5x faster travel is only present during the catch-up phase of geo-fence mode, and travel to the POI is done according to posted speed limits or at walking speed depending on which mode it is in.

With the current implementation of Geofence mode's algorithm, the phone reports traveling to the closest POI (and by the shortest path to that POI), however, more choices and possibilities could be added. For example, given a list of *n* number of POIs, the algorithm could choose one at random to travel to. Or perhaps, there might be a different POI for the day of the week. Additionally, the algorithm could be tweaked to travel between a few POIs and dwell at each for a variable amount of time. All of these options may provide a more realistic appearance, especially when reviewed over the course of weeks or months when the same Geofence configuration is used.

While Geofence mode does work as intended, Two Six observed that it required a bit of an artistic touch and testing to get something that appeared reasonable. If a user was not careful and drew a geofence haphazard, intersecting roads at weird angles or blocking major thoroughfares, IPA would often report an abrupt change in direction to traverse the same road in the opposite direction. We discovered smaller, and carefully drawn geofences provided the best results for realistic travel when observed from a mapping application.

4.3.3 Dither Mode

Dither mode provides the ability for an end user to specify a particular path/route to be reported by IPA when a geo-spatial trigger activates. All Dither mode configurations specify a beginning geo-spatial trigger; once a phone enters this trigger area, it will activate IPA and report the path specified in the configuration. Each point along the path can have a different speed and dwell time associated with it. For example, travel between point A and point B can be 5 kmh, while travel between point B and C can be 50 kmh. Further, the configuration can specify a dwell time of 1 minute at point A, 10 minutes at point B, and 1 hour at point C. Additionally, the travel between points does not need to be on a road network. This allows for flexibility for the end user to appear at certain locations for a specified time period, as well as allows for projected movement within buildings like malls or museums.

When Dither mode reaches the end of its prescribed schedule, it will dwell at the last point waiting for Dither mode to end. The terminating condition for Dither mode occurs when the phone's real and projected location come within 15m of each other. In the case where Dither mode finishes the schedule, then the phone should come within 15m of the end point. However, this can also be done while Dither mode is still projecting/playing the configured schedule. This gives the flexibility for the user to end Dither mode early if desired. An additional capability to Dither mode for future consideration would be to add the catch-up feature to the end of Dither, obviating the need for the user to travel to a specific location.

4.3.4 Orchestration Mode

Orchestration mode is similar to Dither mode in that it allows for a custom schedule/route to be created. However, Orchestration mode was created to allow IPA to be more easily configured to play a schedule for a number of days or weeks. This was ultimately modeled and configured as a series of events and routes between events. This schedule is created using a calendar view within the IPA configuration tools, adding events at different times within the calendar. As different events were added the configuration tools would automatically create a route between two events utilizing open source routing library from Graphhopper [6]. This automatically generated route could also then be adapted/customized by the user if they desired. If the configuration tools detected there was not enough travel time between events, it would warn the user and prevent events from overlapping.

Additionally, Orchestration mode can be configured with 3 different start triggers; geo-spatial, time based, and manual. Like Dither mode, Orchestration can be configured to start when a device enters a specified area. The second method is time-based, whereby IPA will set an alarm for the time of the first event, and when confirmed by the user, IPA will begin to play the schedule. Finally, the manual trigger allows the user to start Orchestration mode at a time of their choosing. As a result of the three different starting triggers, users should be aware that a schedule may not end exactly as displayed in the configuration tools. Specifically, if a schedule is created to start at 9AM one day and end at 9AM the next, it is not necessarily the case that it will end at 9AM. Instead, it should be thought of as a 24 hour schedule that will play when it is triggered, and as such the end time is dependent on the actual time it is triggered.

Once triggered, Orchestration mode behaves mostly the same as Dither mode. IPA will report the location based on the configured schedule, when it is dwelling at locations it will add noise the same way, and ending the schedule works the same as Dither mode, by bringing the phone within 15m of the reported location. The one difference is the addition of a fail-safe mode in Orchestration mode to protect against unexpected crashes or restarts. Since Orchestration mode is meant to run for days or weeks on end, Two Six felt it was necessary to add specific code to prevent reporting the phones actual location unless allowed by the user. So in the event that IPA crashed (or the phone was restarted) while in Orchestration mode, IPA would only report the phone

as being at the first event of the schedule. If IPA was in this fail-safe mode, there would be a notification for the user indicating it, whereby they could go to the location and disable IPA when they were ready.

4.4 Miscellaneous Findings and Observations

Throughout the research conducted in IPA, there were some additional findings and observations that did not prevent IPA from functioning properly but could provide better and more realistic location reporting against closer inspection if they were also generated and reported. First, while IPA is able to report a different GPS coordinate (latitude and longitude), it does not update the bearing, speed, or accuracy of the GPS coordinate. Additionally, IPA does not change any of the reported GPS satellite information that can be requested by applications. On the RF side, both Wifi and cell tower scans from IPA only needed to report the Access Point or Tower identifiers, however, future work should also account for adjusting relative signal strength indicator (RSSI).

Finally, the IPA algorithms are very structured, and schedule driven. Travel between points is done at a constant speed with no acceleration or deceleration, no stops at intersections, no apparent traffic at all. Further, dwell time at locations is fixed and is precisely what was configured. Future enhancements should look to bring speed and dwell time variability into the algorithms at a minimum. Additionally, sources of traffic data (even just historic data) could help make the overall location movement look more realistic.

5 Conclusions

Through the research conducted on the IPA project, Two Six has shown that it is possible to report an alternative location to requesting applications, whereby preserving the privacy of the user. Two Six has also established that the required information currently needed to conceal a phones location with an alternative one, is actually quite low, requiring only modifying the reported GPS latitude and longitude. Further, we demonstrated that it is possible to report alternative locations more robustly, creating realistic movement within a target area by leveraging open source data sets that may stand up to stronger scrutiny. Given the results observed on the phones as well as through analyzing backend data available to us, Two Six believes that this IPA and other similar technology can adequately conceal a phones location from any application on the phone. However, it must be stated, that IPA does nothing to prevent detection of the phone of physical cellular infrastructure or IP based geo-location algorithms.

6 Citations

[1] "Android Trackers," 24th November 2017. [Online]. Available: https://privacylab.yale.edu/press/android- trackers.

[2] J. Hsu, "The Strava heat map and the end of secrets," 29th January 2018. [Online]. Available: https://www.wired.com/story/strava-heat-map-military-bases-fitness-trackers-privacy/.

[3] "OpenStreeMap," [Online]. Available: https://www.openstreetmap.org/.

[4] "WiGLE: Wireless Network Mapping," [Online]. Available: https://wigle.net/.

[5] "MLS - Overview," [Online]. Available: https://location.services.mozilla.com/.

[6] "Open Source - Graphhopper," [Online]. Available: https://www.graphhopper.com/open-source/.

7 List of Acronyms

AOSP – Android Open Source Project

CONOPs – Concept of Operations

GPS – Global Positioning System

IP – Internet Protocol

IPA – Increasing Privacy for Android

Kmh – Kilometers per Hour

NYC – New York City

OS – Operating System

POI – Point of Interest

RF – Radio Frequency

RSSI – Relative Signal Strength Indicator

US – United States

WiFi – Wireless Networking Technology