LocationSafe: Granular Location Privacy for IoT Devices

Paper #9

ABSTRACT

Today, mobile data owners lack consent and control over the release and utilization of their location data. Third party applications continuously process and access location data without data owners granular control and without knowledge of how location data is being used. The proliferation of IoT devices will lead to larger scale abuses of trust.

In this paper we present the first design and implementation of a privacy module built into the GPSD daemon. The GPSD daemon is a low-level GPS interface that runs on GPS enabled devices. The integration of the privacy module ensures that data owners have granular control over the release of their GPS location. We describe the design of our privacy module and then evaluate the performance of private GPS release and demonstrate that strong privacy guarantees can be built into the GPSD daemon itself with minimal to no overhead.

1. INTRODUCTION

Today data owners' personal mobile devices are constantly being tracked and monitored by third party applications without data owners granular consent and control. Data owners' trust is being continuously violated [4].

Data owners have a desire to occasionally share their location data, though desire granular control and approved consent. Third party analysts seek to track data owners continuously. Unfortunately today this tension has resulted in disproportionate control being in favor of the third party analysts.

Recent research has tried to improve user behavior in recognizing permission issues [6], user-defined runtime constraints [13], or tools to help developers identify least-privilege [16].

Additionally, permission managers (e.g., Android and iOS) offer binary permissions to disable or enable location services. However, while this allows data owners to disable location services for applications that do not require location (e.g., Flashlight application) [7], fine grained granularity is still missing. An Android modification called CynagonMod has a module called XPrivacy [18]. XPrivacy enables data owners to configure random or a static location, empty cell ID, blocks geofences from being set, prevents sending NMEA data to application, prevents cell tower updates from being sent to an application, prevents aGPS, returns empty Wi-Fi scans, and disables activity recognition. Ultimately, this provides the data owner control at the application layer.

User applications requesting data of users is a binary permission, either I share my data or I don't. However, sensitive data such as location needs finer control on how accurate and how often the location information is released. Users should be able to control the granularity of their personal data that is released. Users require freedom and control over their own personal data.

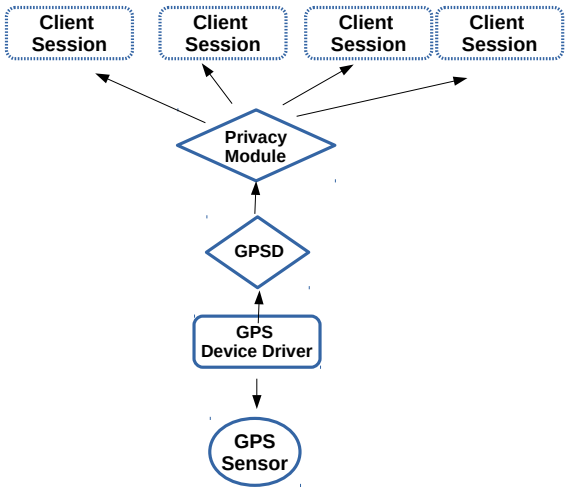

Figure 1: Privatization occurs before data is released to the client application.

However, these approaches discards several important facts: 1) these privacy mechanisms protect at the application layer only and the underlying operating system still has access to all system location APIs 2) granular privacy permission solutions (e.g., XPrivacy) are only for rooted Android phones 3) there is no compromise between third party analyzers and data owners. The expected proliferation of IoT devices will further exacerbate these privacy issues.

In this paper, we present the first (to our knowledge) implementation of a privacy module to GPSD. Figure 1 shows an overview of the flow of queries and responses and demonstrates that the privatization occurs before releasing the data back to the application. The privacy module ensures that all GPS data is released according to the data owner's consent and choice. We demonstrate that appropriate methodologies can be placed which provides strong location privacy guarantees, yet enable analyzers access to privatized location data.

1. A privacy module that integrates into the GPSD software (runs on every GPS enabled device)

2. A granular privacy interface and control to manage location privacy settings (e.g., location coarseness and release frequency)

3. A performant privacy module with minimal overhead

We first describe the architecture and flow of GPSD, we then describe our privatization algorithms, then we describe our integration with GPSD, and finally we evaluate our scheme.

2. RELATED WORK

GPSD is a daemon that network enables the GPS sensor on the majority of mobile embedded systems including Android, iOS, Windows Mobile, UAVs, and driverless cars [9]. On smartphones the network access is limited to localhost applications only (as opposed to remote applications). GPSD enables unfettered access to location data and does not enable or provide any privacy guarantees. LocationSafe provides a privacy module that provides uniform private access across all platforms.

Mobile device permission systems has received attention in the past. Human interaction studies which seek to enhance reader comprehension have been proposed and evaluated [6, 5]. Such systems lack strong and enforacable privacy guarantees. Static analysis tools have been proposed [19]. Though such systems serve only to notify the data owner of privacy breaches and are unable to enforce any privacy runtime guarantees. However, these solutions modify the underlying OS thus making them specific to a single OS or device [20, 18]. Furthermore, these solutions are unable to balance the privacy and utility tradeoff, ultimately resulting a binary approach to privacy.

To guarantee data owner privacy upon the release of data, various mechanisms have been proposed [12, 14, 11, 2, 3]. Differential privacy has emerged as the strongest of these privacy mechanisms [2, 3]. The core idea of differential privacy is to provide strong bounds and guarantees on the privacy leakage when multiple aggregate analytics are run despite the presence or absence of a single data owner from the dataset. This privacy mechanism is provided by adding differentially private noise to the aggregrate answer. As opposed to the originally proposed differentially private mechanism which first collects data in a centralize database and then privatizes the release of the data, LocationSafe immediately privatizes the data at the data source (sensor) in real-time.

3. GOALS AND PROBLEM STATEMENT

We now describe the system goals, performance goals, threat model, and privacy goals of LocationSafe.

3.1 System Goals

There should be well defined and enforced constraints regarding third party application's (apps) access to location data. The data owner should be able to specify the constraints such as how accurate location information should be disclosed and how frequent the location data should be disclosed.

Apps only have access to the privatized data and are unable to directly access GPSD daemon and data. All location data released must be approved by the data owner.

The system should support applications that need real-time access to location data. The privacy policy defines how frequently the application is allowed to receive updates (express in epochs), how accurate the location data may be, and geographical regions as to where the application is allowed to receive location data from.

We use a social network messaging application as an example. The application may want to know which city an individual is in, though pinpoint location information within meter accuracy is not required. The data owner is allowed to define both the radius (e.g., city) that is allowed to be returned as well as the frequency (e.g., say at most every hour).

Ultimately the data owner has final say over how location data and the tradeoff between privacy and utility. The utility has benefits for third party analysts interesting in learning aggregate behavior.

3.2 Performance Goals

The system should scale gracefully as the number of applications connecting to the GPSD daemon increases. Location data will be released within the defined epochs.

3.3 Threat Model

Mobile devices (e.g., smartphones, tablets, wearables) are under the data owner's control. Kernel and underlying OS is vetted and verified (signatures and trusted sources). Focus is not on low level system threats. We assume that the operating system itself is not malicious and provides a mechanism to provide a privacy policy settings manager accessible to the data owner. Secure micro kernels such as seL4 address these issues and are out of scope for this paper. Applications do not have a system exploit (e.g., rootkit) to circumvent the system.

Applications may try to request data more frequently than the defined epoch. LocationSafe will deny such aggressive requests and ensure that data is only released within the defined epoch.

Applications may act as sybils and send false application IDs in order to confuse the GPSD daemon. LocationSafe will treat sybil applications accordingly using data owner defined defaults. Thus, sybil applications may either be receive location data using default privacy configurations or not at all.

3.4 Privacy Goals

Data owners should be able to limit how frequently an application access location data. Data owners should also be able to define fine-grained access to location data. Applications for which the data owner feels the application does not meter level accuracy, the data owner should be allowed to define a radius from which the location value can be returned from. Additionally, for scenarios where fine-grained location is required, the data owner can define a grid system from which potential locations can be returned from.

GPS sensor data is only accessible via GPSD.

4. ARCHITECTURE

Figure 2 depicts the main components GPSD event loop: accepting new client connections, accepting new client sub-

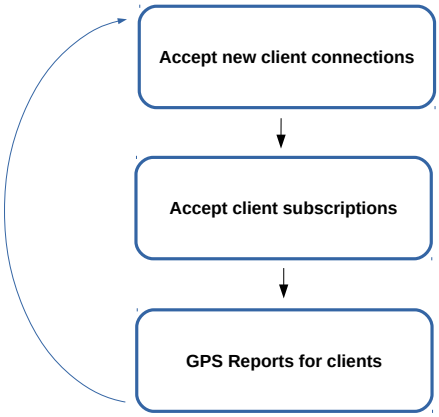

Figure 2: GPSD event loop. Privatization occurs when reporting GPS data to the client.

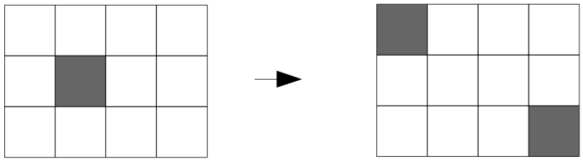

Figure 3: In the grid privatization a single location may randomize to one or many locations. In the example above two locations are returned. However, in the aggregate the analyst is able to estimate the underlying population value without violating individual privacy.

scriptions, and GPS reporting to all subscribed clients. Each client connecting passes in an application identifier which is mapped to a privacy configuration managed by the system. The privacy configuration contains the epoch (how often data is released in milliseconds), differential private ϵ, privatization radius in meters, and the randomized response coin flips. Clients are allowed to pass in a recommended set of privacy parameters, though these are checked against user settings and are not allowed to exceed the privacy threshold defined by the user. In such cases user settings are adhered to.

4.1 Privatization

LocationSafe currently supports two modes of privatization: radius privacy and grid privacy via differential privacy.

The first mechanism is via radius privacy whereby the data owner can specify a radius cover wherein a random point within the defined range is chosen. This approach favors strong privacy at the expense of utility. That is a larger radius grants more privacy though limits the location accuracy.

The second privacy mechanism represents the location space as a grid. The grid can be sized according to the data owner's specification. The current location is placed within the grid. Then leveraging the randomized response method one or many grid locations are returned as seen in Figure 3.

Randomized response [17] was originally created by social scientists as a mechanism to perform a population study over sensitive attributes (such as drug use or certain ethical behaviors). Randomized response allows data owners to locally randomize their truthful answer to analyts' sensitive queries and respond only with the privatized (locally randomized) answer. We utilize randomized response as our privacy mechanism as randomized response satisfies the differential privacy guarantee for individual data owners, it provides the optimal sample complexity for local differential privacy mechanisms [1], and it easily suitable for the location grid type answers we provide.

4.1.1 Mechanism Description

We will now describe how each data owner privatizes their response utilizing the randomized response mechanism. Suppose each data owner has two independently biased coins. Let the first coin flip heads with probability p, and the second coin flip heads with probability q. Without loss of generality, in this paper, heads is represented as "yes" (i.e., 1), and tails is represented as "no" (i.e., 0).

Each data owner flips the first coin. If it comes up heads, the data owner responds truthfully; otherwise, the data owner flips the second coin and reports the result of this second coin flip.

Suppose there are N data owners participating in the population study. Let $\hat{Y}$ represent the total aggregate of "yes" randomized answers. The estimated population with the sensitive attribute Y_A can be computed as:

$$Y_A = \frac{\hat{Y} - (1-p) \times q \times N}{p} \quad (1)$$

The intuition behind randomized response is that it provides "plausible deniability", i.e., any truthful answer can produce a response either "yes" or "no", and data owners retain strong deniability for any answers they respond. If the first coin always comes up heads, there is high utility yet no privacy. Conversely, if the first coin is always tails, there is low utility though strong privacy. It has been shown that by carefully controlling the bias of the two coin flips, one can strike a balance between utility and privacy (Table 4 in [8] and Table I in [10]).

4.1.2 Multiple Sensitive Attributes

While randomized response is an intuitive privacy mechanism for a single location, naturally the question becomes how does one deal with multiple locations, i.e., a grid representation? A host of "polychotomous" mechanisms have been studied and surveyed in the literature [8] using multiple randomizing mechanisms or maximum likelihood estimators [15]. However, it turns out that simply repeating an application of [8] for each grid location turns out to be an "optimal" [15] approach.

Thus, LocationSafe repeats the randomized response mechanism for each grid location. For example, if a traffic analyst wishes to understand the traffic flow of a few key

		Epoch (seconds)		
		5	10	15
# Clients	25	7	12	16
	64	6	11	14

Table 1: Scaling performance of clients receiving a response in specified epoch. Values are averaged across ten iterations.

locations, the traffic analyst issues a query that is a Boolean bit-vector asking each data owner to indicate the location they are at. Then, each data owner performs randomized response for each location and replies with a Boolean bit-vector. The traffic analyst then aggregates and sums the bit-vectors to calculate the number of vehicles at each location.

5. EVALUATION

To evaluate the overhead of the addition of the privacy module to GPSD, we run 25 and 64 clients connecting to GPSD with varying epochs of 5,10,15 seconds as seen in Table 1. The evaluation was run on a laptop running Arch-linux release 2016.06.01 kernel 4.5.4 with two i5 physical cores (four logical) and 12gb ram. GPSD by default has a limit of 64 clients so we stay within this bound.

The results show that minimal overhead is incurred by the privacy module and that clients are able to reasonably receive location updates within the allotted epoch. Even as more clients connect the performance guarantees do not degrade.

6. CONCLUSION

In this paper we present to our knowledge the first software privacy module for GPSD which is a GPS daemon running on the majority of mobile embedded systems today. Data owners are able to express privacy consent and control by enforcing privacy at the lower level of the OS with minimal runtime overhead.

For future work we plan integration with Android and iOS. This will allow us to evaluate the impact and design on location based services.

7. REFERENCES

[1] J. C. Duchi, M. J. Wainwright, and M. I. Jordan. Local privacy and minimax bounds: Sharp rates for probability estimation. In C. J. C. Burges, L. Bottou, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States.*, pages 1529–1537, 2013.

[2] C. Dwork. Differential privacy. In M. Bugliesi, B. Preneel, V. Sassone, and I. Wegener, editors, *Automata, Languages and Programming, 33rd International Colloquium, ICALP 2006, Venice, Italy, July 10-14, 2006, Proceedings, Part II*, volume 4052 of *Lecture Notes in Computer Science*, pages 1–12. Springer, 2006.

[3] C. Dwork, F. McSherry, K. Nissim, and A. Smith. Calibrating noise to sensitivity in private data analysis. In *TCC*, 2006.

[4] Facebook changes story, now says phone location not used to recommend friends. http://fusion.net/story/319712/facebook-now-says-phone-location-not-used-to-recommend-friends/.

[5] A. P. Felt, S. Egelman, and D. Wagner. I've got 99 problems, but vibration ain't one: a survey of smartphone users' concerns. In T. Yu, W. Enck, and X. Jiang, editors, *SPSM'12, Proceedings of the Workshop on Security and Privacy in Smartphones and Mobile Devices, Co-located with CCS 2012, October 19, 2012, Raleigh, NC, USA*, pages 33–44. ACM, 2012.

[6] A. P. Felt, E. Ha, S. Egelman, A. Haney, E. Chin, and D. Wagner. Android permissions: user attention, comprehension, and behavior. In L. F. Cranor, editor, *Symposium On Usable Privacy and Security, SOUPS '12, Washington, DC, USA - July 11 - 13, 2012*, page 3. ACM, 2012.

[7] Android flashlight app tracks users via gps, ftc says hold on. http://www.techrepublic.com/blog/it-security/why-does-an-android-flashlight-app-need-gps-permission/.

[8] J. A. Fox and P. E. Tracy. *Randomized response: a method for sensitive surveys*. Beverly Hills California Sage Publications, 1986.

[9] gpsd - a gps service daemon. http://www.catb.org/gpsd/.

[10] J. Joy, S. Rajwade, and M. Gerla. Participation Cost Estimation: Private Versus Non-Private Study. *ArXiv e-prints*, Apr. 2016.

[11] N. Li, T. Li, and S. Venkatasubramanian. t-closeness: Privacy beyond k-anonymity and l-diversity. In *ICDE*, 2007.

[12] A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkitasubramaniam. l-diversity: Privacy beyond k-anonymity. In L. Liu, A. Reuter, K. Whang, and J. Zhang, editors, *Proceedings of the 22nd International Conference on Data Engineering, ICDE 2006, 3-8 April 2006, Atlanta, GA, USA*, page 24. IEEE Computer Society, 2006.

[13] M. Nauman, S. Khan, and X. Zhang. Apex: extending android permission model and enforcement with user-defined runtime constraints. In D. Feng, D. A. Basin, and P. Liu, editors, *Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security, ASIACCS 2010, Beijing, China, April 13-16, 2010*, pages 328–332. ACM, 2010.

[14] L. Sweeney. k-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(5):557–570, 2002.

[15] A. C. Tamhane. Randomized response techniques for multiple sensitive attributes. *Journal of the American Statistical Association*, 76(376):916–923, 1981.

[16] T. Vidas, N. Christin, and L. Cranor. Curbing Android permission creep. In *Proceedings of the Web 2.0 Security and Privacy 2011 workshop (W2SP 2011)*, Oakland, CA, May 2011.

[17] S. L. Warner. Randomized response: A survey technique for eliminating evasive answer bias. *Journal of the American Statistical Association*, 60(309):63–69, 1965.

[18] Xprivacy. https://github.com/M66B/XPrivacy.

[19] Z. Yang, M. Yang, Y. Zhang, G. Gu, P. Ning, and X. S. Wang. Appintent: analyzing sensitive data transmission in android for privacy leakage detection. In A. Sadeghi, V. D. Gligor, and M. Yung, editors, *2013 ACM SIGSAC Conference on Computer and Communications Security, CCS'13, Berlin, Germany, November 4-8, 2013*, pages 1043–1054. ACM, 2013.

[20] Y. Zhou, X. Zhang, X. Jiang, and V. W. Freeh. Taming information-stealing smartphone applications (on android). In J. M. McCune, B. Balacheff, A. Perrig, A. Sadeghi, M. A. Sasse, and Y. Beres, editors, *Trust and Trustworthy Computing - 4th International Conference, TRUST 2011, Pittsburgh, PA, USA, June 22-24, 2011. Proceedings*, volume 6740 of *Lecture Notes in Computer Science*, pages 93–107. Springer, 2011.