# Hide, Don't Seek: on User-Centric Location Accuracy

Marta Piekarska
Technische Universität Berlin
Berlin, Germany
Email: marta@sec.t-labs.tu-berlin.de

***Abstract*—Location Privacy has been a very attractive topic for the past years. The amount of location blurring, hiding and privacy preserving solutions proposed by the researchers is very big. The number of such services implemented on the market, however, is quite small. Moreover, the ones actually used by the people is close to zero. We see this gap as very troubling. With the spread of devices that will allow for tracking users - the smartwatches, connected medical devices, smartphones and fitness trackers, location privacy will become even more important. When the devices become connected into the Internet of Things, so that it will be hardly possible to escape tracking: already today 70% of users sleep with their phones next to their beds. In our work, we approached the problem from a user-centric perspective, developed our solution together with users and learned what are their expectations towards such a feature.**

***Keywords–mobile privacy; location accuracy; user-centric privacy.***

## I. Introduction

Location based services (LBS) use the information about the geographical position of a mobile device to enhance the quality of experience (QoE) of the user. There are common use cases to which users may be completely oblivious to, for example geotags in images taken by the phone camera. There is a flip side of the LBS. From a privacy perspective exposing the location data to a third-party poses a serious threat. Once the location information leaves a device it may be subject to interception, improper handling, leakage or sale. That causes users' discomfort [1], which has led to development of the European Directive on Privacy and Electronic Communications and decision that location-based services must be permission-based [2]. This means that the end user should opt-in to the service in order to use it.

As recently presented, users identify three discomforts connected to usage of mobile devices: feeling in control, ease of use and ability to take actions [3]. Currently, none of the available mobile OSes introduces any kind of fine grained, flexible control of the location data. In Android one has to accept access to geolocation during installation. In iOS, the user is prompted in real-time, but the only choice she can make is a binary one. Neither of the solutions deals with the concerns found in [3].

We approached the problem from a user-centric perspective and present the results in the paper. We contribute with our work on the following levels:

- Improved location obfuscation based on the grid algorithm.
- Location Privacy preserving solution using a user-centric method
- Very high fliexbility of the solution
- Presentation of user-study and lessons learned from the development of the solution.

The rest of the paper is organized as follows. We start in the next section by evaluating the work done in the field. In Section III, the general solution is presented, which we later improve through the use of a user-centric approach. Results of these consultations and user expectations are described in Section IV. Afterwards we focus on the lessons that can be learned from our work through user-study evaluation and technical analysis. We present the findings and action points that come out from those in Section V. We finish with conclusions in Section VI.

## II. Evaluation of Existing Solutions

**Technological Aspects** Minami and Borisov provide a formal definition of location privacy and suggest a possible solution to the problem in [4]. Andre et al. show another approach where they formalize the intuitive notion of protecting the user's location within a radius r with a level of privacy that depends on r, in [5]. Their work can be generalized to a version of differential privacy. A recent work of Shokri et al. [6] tries to formalize the location–based application, model various location–privacy preserving mechanisms and establish an analytical framework, based on Bayesian inference in Hidden Markov Processes. For more work on privacy preserving systems interested reader should also look into [7]. We have used those works as the background to understand the problem of location obfuscation.

A popular measure of location privacy is the k–anonymity, where we consider a position as hidden if it is indistinguishable from k–1 other users in the same region. First works were done by Gruteser et al [8] where they have proposed to discuss the cloaking region defined as a Quad-tree. Another variant of the same approach was presented in New Casper by Mokbel et al in [9]. Other works introducing the same concept include [10], [11] or [12]. Work of Ahamed et al. [13] attempts to deal with the biggest problem of the k–anonymity - most frameworks use a Location Anonymizer, which is a trusted third-party (TTP) that can define how many people are in a region. Another proposal on how to replace the TTP was given by Ristenpart et al. in [14]. Here, authors suggest using OpenDHT as the third party service for tracking lost and stolen devices. An OpenDHT is a distributed hash table server that has been described in [15]. Any k–anonymity approaches have

 ISBN: 978-1-61208-502-9 

been proven weak and rather useless by Diaz in [16]. After careful analysis of the approaches to location obfuscation we have decided to use the grid method, as the most convenient one and not requiring any TTP. It also did not introduce any computing overhead.

**Apps** More generic permission controlling frameworks have been proposed for Android. Nauman et al. developed a very extensive policy enforcement framework that removes the problem of binary permission granting [17]. Within the scope of their work they give the user a possibility of restraining access to location while preserving the ability of the application to work. Another work in the general field of policy enforcement is CRePE [18]. It is a context–related policy enforcement for Android and can be used to define fine-grained policies. Examples of leveraging the role-based access control include DR BACA [19] and CtRBAC [20]. A different approach is taken in MockDroid [21]. Here authors modify the OS in such a way that the user can "mock" the applications access to the resources. Similar approach has been proposed in AppFence [22]. Zhou et al. propose a flexible, dynamic control of fine–grained permissions in different scenarios with TISSA [23]. Market Place applications that allow the users to block access to location [21] or use fake location [24], are not done on the system level, and suffer from big usability problems.

**Psychology** Research by Fu and Lindqvust [25] gives an understanding of the problem from more psychological perspective. They evaluated how people understand Android's location permissions and how will users' behavior change after we educate them on their true meaning. In [26] Tsai et al. evaluate the peoples' perception of the location sharing cost to benefit ratio by an online survey. They found that respondents feel that the risks of using location-sharing technologies outweigh the advantages. When building a privacy preserving solution, it is often assumed that, in order to improve the users' practices, they should be presented with the possibility of setting a small number of privacy profiles, which can provide basis for configuring individual preferences. According to the study presented in [27], however, this approach can encourage users to share significantly more without a substantial difference in comfort. Based on their three week study, Benisch et al [28], describe a methodology for quantifying the effects, in terms of accuracy and amount of sharing, of privacy-setting types with differing levels of complexity. Other works on the factors impacting users' decision to give away information can be found in [29] and [30]. All of the above gave us understanding of the mindset of the users and led us to designing a system with the choice of per-app setting with various adjustment mechanisms.

## III. General Solution

As presented in Figure 1, we introduce four different ways to deal with an LBS. User, as always, can give to the app his precise position, or decide to turn the location services off. However, we also give him the possibility of meeting privacy half way - by either adjusting the granularity of the position, or completely faking it. Because the feature is not part of the prompt window, but an element of its own in the Settings, user can always go back and change her choice.

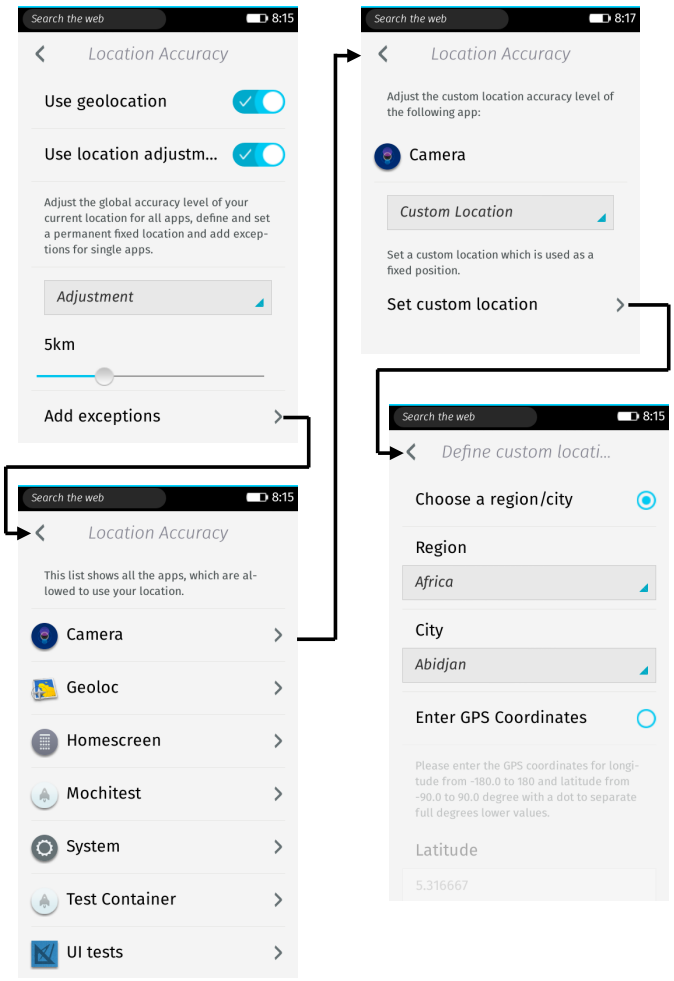


Figure 1. Flow of the Extended Location Settings

*1) Faking the location:* There are many situations in life where we do not want to reveal where we are. Unfortunately, this is not something an LBS will accept: either the application will stop working or it will not even get installed. In addition to that, when one wants to access a service from a location that does not allow for it - certain web pages or shops will only work in a given country. Thus faking your position to the region that is actually acceptable by the app, will extend the functionality. We provide two methods of faking the position. The user can either choose from a predefined set of continents and major cities, where the coordinates are set to the center of the city. The second method is entering position by hand, to allow for precise setting. Figure 2 provides a close–up view of the choice screen.

*2) Adjusting the accuracy:* Going one step towards usability, we have also implemented a mechanism that allows to change the granularity of information shared with the app. As [31] has showed, people have much less concern when only obfuscated location is revealed. We have designed our "Adjustable Location Accuracy" setting, so that it adds necessary noise to the precise position and returns values that are within the limits set by the user for a given app. After obtaining precise location from the available sensors, we use an obfuscation method to provide a certain level of uncertainty.

 ISBN: 978-1-61208-502-9 

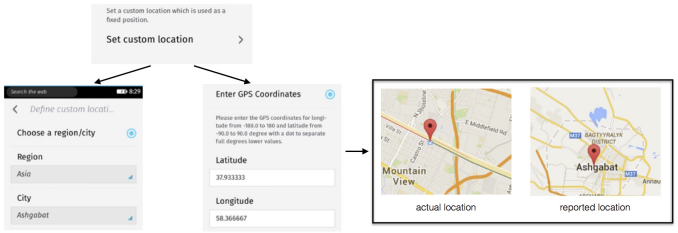


Figure 2. Fixed location setting.

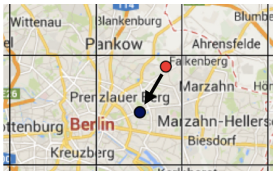


Figure 3. Approximating the position with a grid algorithm

The fuzzing is realized by introducing a grid, where each cell is of the dimensions chosen by the user. The coordinates are used to determine the square in which she currently is. Then we find the middle point of that cell and return that to the application. The visualization of the process is presented in Figure 3. Similar algorithm was first proposed by Micinski et al in [32].

As we deal with a sphere and cartesian coordinates, it has to be taken into account that the size of the grid will change towards the poles. We thus use separate equations to calculate the latitude and longitude. For the latitude, we simplify the calculations by assuming a fixed radius of the earth halfway between the poles and the equator. It is calculated from the radius at 45 degrees North.

First we need to convert the precise latitude into the universal polar stereographic (UPS) coordinate system, so that it accounts for the ellipsoid changes. Next, we find the southern edge of the grid cell, in radians, and add half of a grid cell size to find the center latitude. Lastly, we need to convert the calculation back to degrees, properly wrap it back and return the value in degrees. An ellipsoid with semi–major axis A and semi–minor axis B, has a variable radius at an angle of latitude, $\phi$.

*3) Implementing per-app Settings:* The system is designed to have a single global value, and optionally, a setting per-origin (i.e., per-app). If a geolocation request comes in from an origin that does not have an origin-specific setting, the global setting is used. But this system presents a problem since the existing geolocation system tracked requests by "watch ID" instead of origin. The relationship between origin and watch ID is many-to-one; each origin can have multiple watches set up and running. The geolocation subsystem assigns an unique watch ID for each active geolocation request. Updates from the positioning system come in and then get reported, unchanged, back to the owner of the request. To apply the correct fuzzing behavior, we required a way to get the current fuzzing settings for the origin associated with a given watch ID. We accomplished this by recording the origin in a hash table whenever a new watch ID was generated and clearing the mapping whenever a request was destroyed and the watch ID invalidated.

*4) Implementation of Various Levels of Precision:* For the "precise location" setting, the behavior is exactly the same as before; the location data is sent directly to the requesting origin without modification. The "fixed location" setting is straightforward. Regardless of what geolocation data come from the sensors, we report the user settable fixed location to the requesting code.

For the "no location" setting, we just destroy the location data and report nothing. To maintain expected behavior for the W3C geolocation API, we had to allow app code to request a geolocation watch and get one, but there is no standardized value for "no location" so we chose to just prevent any data from getting reported.

The most difficult setting to implement is the approximate, fuzzed location. We created a class to encapsulate the fuzzing behavior. We implemented the WGS84 geodetic system for the earth ellipsoid. The constants chosen come from [33]. To simplify the calculations we assumed a fixed radius of the earth at 45 degrees north latitude, and the longitudinal radius was based on a given longitude.

There are attacks against the algorithm we have used. When the user is moving, it can be observed over time how he changes the cells, thus the probability of revealing his location from a square, drops to a linear one. Moreover, if the attacker will take the road map of a given region and compares it with the data collected, the cloaking ratio drops to $\frac{1}{No.ofroads}$. To reduce the probability of succeeding, we introduce a jitter when the device is moving. This will not remove the possibility of performing the attack. It will, however, introduce additional level of uncertainty. Based on the speed we include a random delay with which the position is being changed.

## IV. User Expectation

After creating a simple prototype of a location preserving solution that would allow for adjusting location accuracy, we have worked with a group of "every day experts" – users who are not technically savvy towards improving and building a solution that would meet their expectation. We used the user-centric approach discussed in [3]. From that we have found that we should provide users with ability to set different levels of settings for different applications, add contextual dependencies and create profiles. We describe those concepts below.

### A. Application–transparent obfuscation

On the first level a user is asked to define the global, system-wide setting, that will be the default choice for any app. In addition, one can create a list of applications that will be subject to different adjustment.

The apps can be grouped by vendor, sorted alphabetically or by the trust level. By the latter we refer to the definition of "web", "privileged", and "certified". Moreover, the user can just search for the applications by name. We give the users choice of such filtering to help them with making the decision on how much do they trust the applications. By allowing to group apps by vendor name user can see which of them may potentially collaborate. Sorting based on system's definition of

 ISBN: 978-1-61208-502-9 

trust gives intuition about the risk of potential maliciousness of an app. Additional ways to improve that mechanism would be adjusting the location granularity based on the application type. However, Firefox OS currently does not provide information that would allow for such grouping.

### B. Adding the Context to Decision Making

In addition to per-app setting we introduce another level of definition. Following the concepts presented in other fields, our solution gives the user a choice of setting both temporal and spatial intervals where chosen level of adjustment will be set. This means that now the user can define time span and space where the location granularity will be changed. This solution is useful when entering the area or time slot of increased privacy - being in a hospital or during the night.

### C. Creating Profiles

In addition to above–mentioned extensions we also decided to include user profiles. Following the conclusion that there are situations where one might want to switch into a stealthy mode, or quite the opposite - for the time being needs all applications to access everything - we allow users to save their settings into profiles. This way they can adjust the granularity ahead of time and activate the mode when needed. Such setting, of course, can be time consuming and tiresome. It is however an option, and not a requirement, and can be done at any given point in time. The profiles can also be activated based on temporal and spatial information.

## V. Lessons Learned

One of the biggest contributions of this work is a summary of lessons learned while implementing the design with a user-centric approach and setting out clear action points how to improve the location protecting solutions.

### A. User-study Evaluation

The scope of the user-study evaluation was to verify the impact of our approach. Results presented here only reflect the evaluation of the solution. Additional findings, connected to what where the particular choices users made, how did they differentiate the applications, and how often did they change their configuration, will be further analyzed and presented in a later publication.

*1) Methodology:* The group consisted of 30 participants, 40% males and 60% females fairly equally distributed between 17 and 60 years old. Their educational background varied: 36,7% graduated from a collage, 30% were still in college, 20% had a higher education level like a master degree. The remaining group had either a high school diploma, or finished their education one level before that. These show, that there was a bias towards a higher education levels. The participants were distributed between all kinds of jobs: students, graphic designers, secretaries, clerks, lawyer assistants, project managers, etc. None of the participants have ever used Firefox OS, however all of them previously had a smartphone.

Each participant was invited to the lab, where they were handed the phone, and asked to use it as their primary device. They were not informed about the precise purpose of the study, but we have told them that their behavior would be monitored. Prior to the study, we have obtained their approval, promised the results would be anonymized and only used for research. We incorporated into the typical First-Time-Use run additional screens about the privacy features included in the modified system. The Adjustable Location Accuracy was described with the set of screens presented in Figure 4. We asked participants to use the Firefox OS phone for a week. We monitored and made sure that they were using the devices during the study.

*2) Survey:* After the test period finished we invited the participants back to the lab. We asked them to give their opinions on following statements:

1) I am satisfied with the Adjustable Location Accuracy.
2) I feel that using Adjustable Location Accuracy slows down my phone.
3) I feel that Adjustable Location Accuracy is a feature that was hard to use.
4) How often did you set up the Adjustable Location Accuracy?
5) Name setting that you found most useful.

For the first three questions participants were asked to choose one out of seven possible answers. We have based the choice on the Likert scale: 1 - Strongly disagree, 2 - Disagree, 3 - Somehow disagree, 4 - Neither agree nor disagree, 5 - Somehow agree, 6 - Agree, 7 - Strongly agree [34]. Question number 4 had the following answers possible: 1 - I did not use the feature at all, 2 - I have tried to setup the feature and resigned, 3 - I have setup the feature once, 4 - I have setup the feature repeatedly. In this question we asked the users to justify their answers in free–text response. Last question was multiple choice, limited to 3, with the following possibilities: 1 - I did not find Adjustable Location Accuracy useful at all, 2 - No location, 3 - Precise Location, 4 - Custom Location, 5 - Adjustment, 6 - Per-application settings, 7 - Global setting, 8 - Privacy Profiles. We also left space for free-text comments.

*3) Results:* First question was measuring the general satisfaction with the solution. 80% of the participants chose answer 6 and 7, which means they were highly satisfied with the functions provided, 13% chose “Somehow agree”, and 7% were ambiguous about it (answer 4). None of the users reported lack of satisfaction. For the second question we have seen only answers that did not suggest any delays: 83% said that they strongly disagree with the statement, while 17% chose option 2. With the question number three we wanted to verify what was the perception of out tool. While most users said that they did not find the solution hard (40% Strongly disagree, 26.7% Disagree, 26.7% Neither Agree nor Disagree, 6.6% Somehow agree), in the comments section they have often mentioned that the process of setting up the tool was time consuming. However, when looking at the answers to question 4 we noticed that most users set up the process only once (20 people). 8 participants chose answer number 4, while only 2 participants did not use the feature at all. They argued they “did not feel like it changed anything”. We did not verify if this was due to bad explanation in the Guided Tour or different privacy expectation. Lastly, the users chose mostly the “Privacy Profiles”(83.3%), “Custom Location” (76.7%), “Per application settings” (66.7%). These were followed by “Global Settings” (40%).

 ISBN: 978-1-61208-502-9 

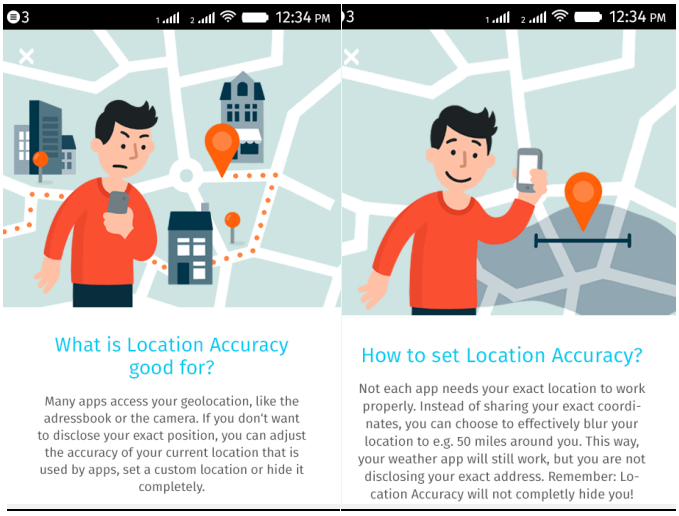


Figure 4. The first two wireframes presented to the participants of the user study performed to evaluate the adaptation of our framework.

Overall we saw that participants were happy to use the feature and did not notice any performance problems. We have concluded that users are happy to adapt Privacy Enhancing Technologies (PETs), as long as they do not have to give up usability. Because PETs do not offer users more features or a better experience, the best way to drive adoption is by making them both transparent and automatic. There is more to be done on the topic of user studies and user-perception of location accuracy. We plan to deeply investigate the topic, as the results to the last question were very interesting to us. We feel that our solution would benefit from an automated setting based on recommendations made by trusted people. However, this might not improve the privacy, as [27] suggests.

### B. Technical Analysis

The location obfuscation does not include any overhead compared to lack of the feature. However to ensure higher impact, it requires to introduce some kind of IP randomization, which we will further discuss in Section V-C.

Our solution does not cause the apps to break and even enhances functionality. Some applications require the user to grant permission to access the geolocation data from the device. By reporting no location, the app receives access but doesn't get any location data that is useful for tracking the user. This empowers the user to gain access to the app without compromising their privacy. This is possible, because applications usually set up a callback to handle the location information once it is obtained from the system. When the user selects "no location", no data is returned and the callbacks never get called. If the application is designed to asynchronously handle location information updates, it usually continues to work and simply uses the IP-based position. The application will typically have a hierarchy of geolocation sources and the API based is highest in the rank. Thus with other settings, like "Adjustment", or "Custom Location" it will just accept the data returned by the Geolocation API.

### C. Fighting IP-based Location

The biggest problem of any location obfuscation solution is the IP-based attack. We propose to fight against it by using a Tor-based solution [35]. In order to do that, some changes in the current implementation of Tor need to be done: the exit nodes should comply with the region chosen by the user location. Assume there are two web sites, A.com and B.com, that both use tracking cookies/images from C.com. If the user visits A.com with Tor and location obfuscation and B.com without Tor and location obfuscation, then C.com's tracking will see two different IP's and locations for the user. If the user then visits a third site with C.com tracking bits on it, then C.com will see the same IP and location data it saw when the user visited B.com. C.com can develop a statistical model for the location and IP of the user that will discount the masked IP and fuzzed location reported when accessing A.com. This problem can be addressed by developing a cookie management method for obfuscation techniques. Currently we believe that the best way this attack can be defeated is by applying the cookie policies found in the Tor Browser Bundle (TBB) and ensuring that all the traffic is forwarded through the Tor. We strongly encourage better investigation how Tor technologies can be incorporated into industry solutions.

### D. W3C standard improvements

The W3C geolocation API takes into consideration the impact on user privacy in Section 4 of the specification [36]. It states that "a conforming implementation of [the] specification must provide a mechanism that protects the user's privacy...". They do not, however, specify how this should be done. Our implementation attempts to further improve basic geolocation with user selectable behavior of the geolocation discovery and reporting mechanism. Moreover we introduce the possibility of setting a custom chosen, fixed position. We believe that it should be an enhancement of the specification to introduce both the "no location" and "fixed position" choices. We do not suggest working on the fuzzing mechanism, as this has not been proven to be secure.

In case of the "custom location" there should be a requirement to allow the user to set the position to a defined value. This would comply with the request of respecting privacy that is not currently enforced. For the "no location" we would like to introduce a constant value. When a user selects the behavior in our implementation, we do not execute any position callbacks to the client application. In some cases, web apps interpret the lack of a callback as an error condition. This would not be the case if a constant was present that would define what "no location" is. Then we could execute the callback with that value and avoid the incorrect error condition. We propose that "no location" be represented by a position value of NaN for the coordinate latitude, longitude, and accuracy as well as a timestamp value of 0.

### E. User-centric Lessons

Working together with the users from the very beginning gave us a good understanding of what people expect of a location obfuscation solution. As with other privacy enhancing technologies, user expect privacy preserving location based services to run in the background and somehow "intuitively

 ISBN: 978-1-61208-502-9 

guess" what would they want to reveal in any given moment. Consent to share location data with an app heavily depends on the context, be it spacial or temporal as presented in [37]. Defining static polices does not conform those requirements for the system. Thus when granting access to location not only should we ask the question of "if" but also "in what circumstances". In addition, extending this by adding per-app settings allows for further flexibility, while introducing profiles makes the management slightly easier. However the more flexible and adjustable the solution is the harder it gets to build a user interface for it. We definitely see this as a challenge to be further explored and investigated.

## VI. Conclusions

In our work towards user–centric privacy, we developed a tool that is a compromise between hiding and usability. With various levels of protection, ranging from turning the LBS off, through providing a fake location, adjusted accuracy all the way to precise position. By including the Guided Tour that explains the feature we educate the user on threats connected to revealing his data and give him opportunity to protect it according to his needs and concerns. We have developed our code on Firefox OS, and tested it on physical devices. The overhead is unnoticeable by the users, as we have presented. Moreover, our user study showed that people do not feel discomfort while adjusting the settings initially. They rather report it is a good way to learn which apps access their location data .We would like to investigate more carefully how can we comply with the six goals suggested publication of Luo and Hengartner [38]. We would also want to analyze a very different concept described by Puttaswamy and Zhao [39] - where the external servers are treated as simple encrypted data stores. This removes the problem of leaking any data to anyone but the parties directly involved. We think that our solution could greatly benefit from this addition. The proposed scheme is a solution to the concerns users have around the location privacy. It is flexible, which gives good QoE. The idea of adjusting location on per–app basis, takes into account the context - users will be willing to share different things with different applications. It is adjustable straight in the settings, so that depending on the situations they can also change their choice. By grouping the apps with respect to their vendor, we give the overview of which apps may be collaborating with each other, which has not yet been seen as an attack vector. The elements of context–awareness and pre-setting profiles make it even more user–centric.

We have identified several action points for the academia and the industry. First is better support of user-centric development of privacy tools in general, and location privacy preserving solutions especially. We strongly believe that this is the only way to get the adaptation rate of PETs higher. Second is improving the current W3C geolocation standard by introducing the "no location" solution. Additionally the geofencing document should be designed with grater care. We see a problem in the fact that the whole drafted before the actual Privacy section was written. Taking into account how sensitive already is the location data, and what the API would allow for it is troubling. The great lesson learned from working together with the users on privacy enhancing technologies is seeing how they care about their data, and protecting it, however they seek solutions that are transparent and intuitive. We believe that it is the role of the standardization bodies and academia to work together towards creating such standards.

## References


[1] S. Patil, G. Norcie, A. Kapadia, and A. Lee, ""check out where i am!": Location-sharing motivations, preferences, and practices," in *CHI '12 Extended Abstracts on Human Factors in Computing Systems*.

[2] "Directive on privacy and electronic communications," European Parliament, Tech. Rep., 12 July 2002.

[3] M. Piekarska, Y. Zhou, D. Strohmeier, and A. Raake, "Because we care: Privacy dashboard on firefox os," in *Web 2.0 Security and Privacy 2015, S&P IEEE*.

[4] K. Minami and N. Borisov, "Protecting location privacy against inference attacks," in *Proceedings of the 9th Annual ACM Workshop on Privacy in the Electronic Society*, ser. WPES '10.

[5] M. E. Andrés, N. E. Bordenabe, K. Chatzikokolakis, and C. Palamidessi, "Geo-indistinguishability: Differential privacy for location-based systems," *CoRR*, 2012.

[6] R. Shokri, G. Theodorakopoulos, J.-Y. Le Boudec, and J.-P. Hubaux, "Quantifying location privacy," in *Security and Privacy (SP), 2011 IEEE Symposium on*, 2011.

[7] M. Herrmann, C. Troncoso, C. Diaz, and B. Preneel, "Optimal sporadic location privacy preserving systems in presence of bandwidth constraints," in *Proceedings of the 12th ACM Workshop on Workshop on Privacy in the Electronic Society*, ser. WPES '13.

[8] M. Gruteser and D. Grunwald, "Anonymous usage of location-based services through spatial and temporal cloaking," in *Proceedings of the 1st International Conference on Mobile Systems, Applications and Services*, ser. MobiSys '03.

[9] M. F. Mokbel, C.-Y. Chow, and W. G. Aref, "The new casper: Query processing for location services without compromising privacy," in *Proceedings of the 32Nd International Conference on Very Large Data Bases*, ser. VLDB '06.

[10] M. Jang, M. Yoon, H.-i. Kim, and J.-W. Chang, "A privacy-aware location cloaking technique reducing bandwidth consumption in location-based services," in *Proceedings of the Third ACM SIGSPATIAL International Workshop on Querying and Mining Uncertain Spatio-Temporal Data*, ser. QUeST '12.

[11] H. Elmeleegy, M. Ouzzani, A. Elmagarmid, and A. Abusalah, "Preserving privacy and fairness in peer-to-peer data integration," in *Proceedings of the 2010 ACM SIGMOD International Conference on Management of Data*.

[12] A. Solanas, F. Sebé, and J. Domingo-Ferrer, "Micro-aggregation-based heuristics for p-sensitive k-anonymity: One step beyond," in *Proceedings of the 2008 International Workshop on Privacy and Anonymity in Information Society*.

[13] S. I. Ahamed, C. S. Hasan, M. M. Haque, and M. O. Gani, "Towards ttp-free lightweight solution for location privacy using location-based anonymity prediction," in *Proceedings of the 2011 ACM Symposium on Research in Applied Computation*.

[14] T. Ristenpart, G. Maganis, A. Krishnamurthy, and T. Kohno, "Privacy-preserving location tracking of lost or stolen devices: Cryptographic techniques and replacing trusted third parties with dhts," in *Proceedings of the 17th Conference on Security Symposium*, ser. SS'08.

[15] S. Rhea, B. Godfrey, B. Karp, J. Kubiatowicz, S. Ratnasamy, S. Shenker, I. Stoica, and H. Yu, "Opendht: A public dht service and its uses," 2005.

[16] C. Diaz, "A closer look at cloaking techniques for location privacy," in *Katholieke Universiteit Leuven*, 2010.

[17] M. Nauman, S. Khan, and X. Zhang, "Apex: Extending android permission model and enforcement with user-defined runtime constraints," ser. ASIACCS '10.

[18] M. Conti, V. T. N. Nguyen, and B. Crispo, "Crepe: Context-related policy enforcement for android," ser. ISC'10.

[19] F. Rohrer, Y. Zhang, L. Chitkushev, and T. Zlateva, "Dr baca: Dynamic role based access control for android," ser. ACSAC '13.

[20] T. T. W. Yee and N. Thein, "Leveraging access control mechanism of android smartphone using context-related role-based access control model," in *Networked Computing and Advanced Information Management (NCM), 2011 7th International Conference on*.


 ISBN: 978-1-61208-502-9 

[21] A. R. Beresford, A. Rice, N. Skehin, and R. Sohan, "Mockdroid: Trading privacy for application functionality on smartphones," ser. HotMobile '11.

[22] P. Hornyack, S. Han, J. Jung, S. Schechter, and D. Wetherall, "These aren't the droids you're looking for: Retrofitting android to protect data from imperious applications," ser. CCS '11.

[23] Y. Zhou, X. Zhang, X. Jiang, and V. W. Freeh, "Taming information-stealing smartphone applications (on android)," ser. TRUST'11.

[24] "http://www.placemask.com/products."

[25] H. Fu and J. Lindqvist, "General area or approximate location?: How people understand location permissions," in *Proceedings of the 13th Workshop on Privacy in the Electronic Society*, ser. WPES '14.

[26] J. Y. Tsai, P. G. Kelley, L. F. Cranor, and N. Sadeh, "Location-sharing technologies: Privacy risks and controls," in *A Journal of Law and Policy for the Information Society 6(2)*, 2010.

[27] S. Wilson, J. Cranshaw, N. Sadeh, A. Acquisti, L. F. Cranor, J. Springfield, S. Y. Jeong, and A. Balasubramanian, "Privacy manipulation and acclimation in a location sharing application," in *Proceedings of the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing*.

[28] M. Benisch, P. G. Kelley, N. Sadeh, and L. F. Cranor, "Capturing location-privacy preferences: Quantifying accuracy and user-burden tradeoffs," *Personal Ubiquitous Comput.*

[29] S. Patil, Y. Le Gall, A. J. Lee, and A. Kapadia, "My privacy policy: Exploring end-user specification of free-form location access rules," in *Proceedings of the 16th International Conference on Financial Cryptography and Data Security*.

[30] J. Y. Tsai, P. Kelley, P. Drielsma, L. F. Cranor, J. Hong, and N. Sadeh, "Who's viewed you?: The impact of feedback in a mobile location-sharing application," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2009.

[31] K. Fawaz and K. G. Shin, "Location privacy protection for smartphone users," in *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*.

[32] K. Micinski, P. Phelps, and J. S. Foster, "An Empirical Study of Location Truncation on Android," in *Mobile Security Technologies (MoST) 2013*.

[33] N. G.-I. Agency, *World Geodetic System 1984*.

[34] I. Allen and C. A. Seaman, "Likert scales and data analyses,," *Quality Progress, vol. 40, no. 7*, 2007.

[35] "Tor project." [Online]. Available: www.torproject.org

[36] W3C, "Geo api specification." [Online]. Available: http://dev.w3.org/geo/api/spec-source.html

[37] S. Consolvo, I. E. Smith, T. Matthews, A. LaMarca, J. Tabert, and P. Powledge, "Location disclosure to social relations: Why, when, & what people want to share," ser. CHI '05.

[38] W. Luo and U. Hengartner, "Proving your location without giving up your privacy," ser. HotMobile '10.

[39] K. P. N. Puttaswamy and B. Y. Zhao, "Preserving privacy in location-based mobile social applications," ser. HotMobile '10.

 ISBN: 978-1-61208-502-9 