# Location Guard: location privacy for the rest of us

Konstantinos Chatzikokolakis
CNRS, LIX, École Polytechnique
France

Catuscia Palamidessi
Inria, LIX, École Polytechnique
France

Marco Stronati
LIX, École Polytechnique
France

## ABSTRACT

Location Guard is an open source [1] web browser extension that provides location privacy when using the HTML5 geolocation API. The privacy notion enforced by Location Guard is geo-indistinguishability, a novel definition inspired by differential privacy. The extension has reached considerable popularity since its release, covering Chrome, Firefox and Opera browsers, and more recently moving to mobile devices with Firefox for Android. The next step for the project is to incorporate ideas from the recently proposed Elastic Mechanism and adapt the level of protection based on the actual location of the user. The talk will present the desktop and mobile versions of Location Guard, as well as its experimental elastic variant, together with what we learned on *usable privacy* from our user base.

## 1. MOTIVATION

Privacy in geolocation has been a prolific research area in the last years, reflecting the growing importance of the problem, mostly due to the pervasiveness of mobile devices. Despite the many theoretical advancements in the area,there is still a lack of practical options for the average user to protect her privacy while using a Location Based Service. Geo-indistinguishability [4], a recently proposed definition of location privacy, has both the advantage of providing a formal and user independent privacy guarantee, and at the same time a simple and efficient mechanism to achieve it. Furthermore the mechanism can be easily configured with a single parameter that provides a simple intuition of the trade off between privacy and utility. For these reason geo-indistinguishability is an ideal candidate to implement in a tool aimed at non-technical users.

## 2. A WEB BROWSER EXTENSION

The Web browser is by far the most used interface between users and privacy sensitive services nowadays; its growth in popularity as a *platform* to develop applications makes it an ideal target for a privacy preserving tool. Browser extensions, allowed to run code with limited privileges in order to offer new functionalities – e.g. ad-blockers, search-engines – are becoming increasingly popular. For our purposes it is possible to intercept calls to the Geolocation API, sanitize the original location provided by the browser, then return a private version to the calling application in a transparent way. A browser extensions allows to incorporate privacy in a great number of services, in a way that is familiar and easy to install.

## 3. DESKTOP AND MOBILE

Despite the fact that location privacy is considered especially important in mobile devices, the accuracy of modern wifi-based location providers is detrimental also to desktop users: the position of a wifi connected laptop can often be determined more accurately than a mobile phone with GPS. Users awareness of the problem is shown by the popularity of the first versions of Location Guard, that were limited to desktop browsers.

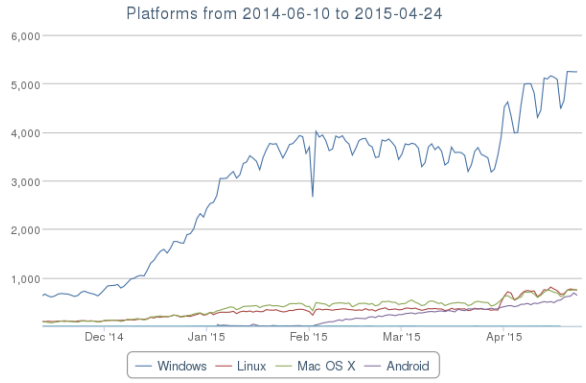


**Figure 1: Firefox usage statistics by platform, including mobile.**

Since release 1.2.0 (February 2015) Location Guard runs on Firefox for Android, currently the only mobile browser supporting extensions, and its mobile user base has been growing rapidly. Supporting mobile devices is crucial since they typically follow all users' movements, while mobile-optimized websites ask for user's location increasingly often.

Although on smartphones native apps are the most popular way to interact with online services, the growing popularity of web applications, to contrast the rampant fragmentation of mobile development, promise a larger coverage of services for Location Guard in the near future. Furthermore Mozilla announced a new port of Firefox to iOS in the next year thus covering the other half of the mobile space.

## 4. OPERATION

Every web application runs in a separate environment and can access privileged information, such as the user's location, through a JavaScript API provided by the browser. A browser extension has the ability to run JavaScript code with higher privileges than a normal page and, among other things, to modify the content of any web page. When a page is loaded and before any other code is executed, Location Guard injects a small snippet of JavaScript that redefines `geolocation.getCurrentPosition`, the main function provided by the Geolocation API to retrieve the current position. When the rest of the page code runs and tries to access this function, it gets intercepted by Location Guard, which in turn obtains the real

location from the browser, sanitizes it and returns it to the page.

The location is sanitized through the use of random noise drawn from a Planar Laplace distribution. The amount of noise added can be configured easily with a single parameter, the privacy *level*. Location guard provides three predefined levels {high,medium,low} and the user is also free to pick any other value. Additionally the privacy level can be adjusted per domain, so that different protection can be applied to different services: a larger amount of noise can be added to a weather service as opposed to a point of interest search engine.

An advantage of geo-indistinguishability is that it is relatively intuitive to explain to the user the effect of changing the levels on privacy and utility. For a certain privacy level we can compute two radiuses $r_p$ and $r_u$, respectively the radius of privacy protection and of utility. $r_p$ is the area of locations highly indistinguishable from the actual one, i.e. all locations producing the same sanitized one with similar probabilities. $r_u$ is the area in which the reported location lies with high probability, thus giving an idea of the utility that the user can expect. Both these radiuses can be easily plotted on a map to give the user a direct impression of privacy and utility, according to the level of protection chosen.

Apart from sanitizing the real location, Location Guard supports reporting a *fixed* predefined location, which offers perfect privacy at the cost of very low utility.

## 5. ONGOING WORK

One shortcoming of standard geo-indistinguishability is that the privacy level has to be fixed independently of the user location. So for example once set to have a protection in a radius of 200m, that is sufficient in a dense urban environment, the same protection will be provided when the user moves outside the city, possibly in sparsely populated area. The problem is described in mode depth in [5], where a solution is proposed in the form of an elastic mechanism that adapts to semantic characteristics of each location, such as population and presence of POIs. However, the extreme flexibility of this mechanism, that can change its behavior for locations just 100 meters apart, comes with the cost of a heavy phase of pre-processing to build its semantic map, which is not suitable for Location Guard.

What we would need is a simplified version of the elastic mechanism where the noise level is adapted to large areas, small enough to distinguish a park from a residential area, but still easily computable. In order to build this set of tiles, Location Guard can query a number of online geographical services, to obtain a set of geographical polygons together with a quantitative description of the amount of privacy they provide (an equivalent of the privacy mass used in the elastic mechanism). This dataset should cover an area large enough to contain most of the user usual movement and it can easily reach a few tens of kilometers while retaining a small size. Once this small dataset is build, we have a mapping from polygons to their privacy mass. We can now use it to define a function $l$ that for each location, finds the containing polygon and returns a privacy level adapted to the privacy mass provided by the polygon.

The mechanism described above, despite achieving the flexible behavior we needed, does not satisfy geo-indistinguishability. It is enough to notice that the level of protection, a public information of the mechanism, depends on the current location of the user, which is sensitive. In order to solve this problem we need to make $l$ itself differentially private; a simple way to do it is to first sanitize the current location with a fixed privacy level and then feed it to $l$. Post processing a sanitized location does not pose any threat to privacy and would allow the mechanism to reduce sharply the amount of noise added to location in very private area.

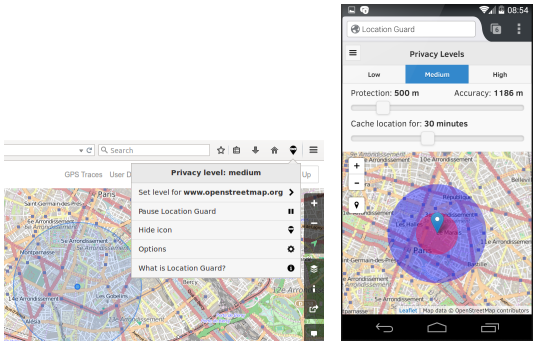


**Figure 2: Left: Main menu on desktop. Right: Privacy level configuration on Android, $r_u$ in purple and $r_p$ in pink.**

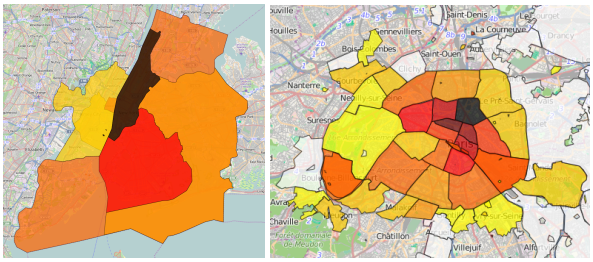

**Figure 3: Polygons computed for New York and Paris**

We are currently evaluating the use of two online services to build the dataset. In order to extract the geographic polygons of administrative areas and natural features (such as lakes and rivers) we are using OverPass Turbo (overpass-turbo.eu), that provides a simple API to query the OpenStreetMap database. Once obtained the polygons we use again Overpass Turbo to query for POIs densities and DBpedia (dbpedia.org) to obtain population densities from Wikipedia; this values are combined in a privacy mass measure of each polygon. Preliminary results are shown in Figure 3.

## 6. ADOPTION AND CONCLUSIONS

As of April 2015 Location Guard counts 8,849 active users in Google Chrome, 7,084 in Mozilla Firefox (including Android) and 3,086 downloads in Opera. Adoption has been mainly through the browser extension stores, as well as through technology blogs covering Location Guard [2, 3]. HotPETs'15 would be the first venue where Location Guard is officially presented and it would provide a great opportunity to increase the visibility of the project and get feedback from the privacy community.

## 7. REFERENCES


[1] https://github.com/chatziko/location-guard.

[2] http://www.ghacks.net/2014/12/01/change-the-default-geolocation-in-firefox-using-location-guard/.

[3] http://korben.info/geolocalisation-restez-maitre-de-votre-situation.html.

[4] M. E. Andrés, N. E. Bordenabe, K. Chatzikokolakis, and C. Palamidessi. Geo-indistinguishability: differential privacy for location-based systems. In *Proc. of CCS*, pages 901–914. ACM, 2013.

[5] K. Chatzikokolakis, C. Palamidessi, and M. Stronati. Constructing elastic distinguishability metrics for location privacy. *CoRR*, abs/1503.00756, 2015.