

# Location Privacy via Geo-Indistinguishability

Konstantinos Chatzikokolakis<sup>1,2</sup>, Catuscia Palamidessi<sup>2,3</sup>(✉),  
and Marco Stronati<sup>2</sup>

<sup>1</sup> CNRS, Paris, France

<sup>2</sup> LIX, École Polytechnique, Rocquencourt, France

<sup>3</sup> INRIA, Paris, France

`catuscia@lix.polytechnique.fr`

**Abstract.** In this paper we report on the ongoing research of our team Comète on location privacy. In particular, we focus on the problem of protecting the privacy of the user when dealing with location-based services. The starting point of our approach is the principle of geo-indistinguishability, a formal notion of privacy that protects the user’s exact location, while allowing approximate information – typically needed to obtain a certain desired service – to be released. Then, we discuss the problem that raise in the case of traces, when the user makes consecutive uses of the location based system, while moving along a path: since the points of a trace are correlated, a simple repetition of the mechanism would cause a rapid decrease of the level of privacy. We then show a method to limit such degradation, based on the idea of predicting a point from previously reported points, instead of generating a new noisy point. Finally, we discuss a method to make our mechanism more flexible over space: we start from the observation that space is not uniform from the point of view of location hiding, and we propose an approach to adapt the level of privacy to each zone.

## 1 Introduction

In recent years, the increasing availability of location information about individuals has led to a growing use of systems that record and process location data, generally referred to as “location-based systems”. Examples of these systems include Location Based Services (LBSs), location-data mining algorithms to determine points of interest, and location-based machine learning algorithms to predict traffic patterns.

While location-based systems have demonstrated to provide enormous benefits to individuals and society, the growing exposure of users’ location information raises important privacy issues. First of all, location information itself may be considered as sensitive. Furthermore, it can be easily linked to a variety of other information that an individual usually wishes to protect: by collecting and processing accurate location data on a regular basis, it is possible to infer an individual’s home or work location, sexual preferences, political views, religious inclinations, etc.

It is therefore important to design and implement methods for protecting the user's privacy while preserving the utility and the dependability of location data for their use in location-based systems. In this paper, we report on the research of the INRIA Comète team on this field.

A characteristics of our approach is that we focus on the problem of *protecting the user's location, rather than the user's anonymity*. The latter is based on the idea of hiding the association between the user's location data and his name. However, there have been several examples of attacks showing that anonymity is not sufficient to protect the user: in the large majority of cases, location data can be re-identified by using correlated information.

Furthermore, we focus on methods that provide privacy guarantees which are (a) *based on solid mathematical basis*, (b) *independent from the adversary side information*, and (c) *robust with respect to composition of attacks*.

Our approach is based on the notion of *geo-indistinguishability*, which is a property similar to that of *differential privacy* [8]. Basically, the idea is to obfuscate the real location by reporting an approximate one, using some random noise. The idea is that from the reported location, the attacker may be able to make a good guess of the area where the user is actually located, but it should not be able to make a good guess of the exact location of the user within this area. This mechanism can be implemented by using a noise with a Laplacian distribution, that is a negative exponential with respect to the distance from the real location, like in the case of differential privacy. This method provides a good level of robustness with respect to composition of attacks, in that the level of privacy decreases in a controlled way (linearly).

When the user makes several repeated applications of the mechanism from related points (typically in the case of a trace), however, even a linear decrease of the level of privacy poses a tall too high to the privacy level. To address this problem, we propose a *predictive mechanism*, which avoids the application of the mechanism when a new (noisy) point can be derived from the previous ones.

Finally, we consider the problem that raises when the space is not uniform with respect to the hiding value: the point is that in different zones the number of locations where the user could be located may vary a lot, and as a consequence these zones should have a different privacy parameter. We address this problem by proposing an *elastic mechanism*, which is based on a notion of distance adapted to the different zones.

## 1.1 Related Work

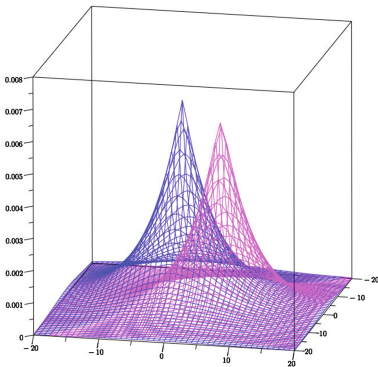
Most location privacy mechanisms proposed in the literature involve obfuscation of the real location. The simplest methods are those based on variants of the *cloaking technique*, which consists in hiding the real location within a *region of possible locations*, for instance by reporting the area around the real location, or by using dummy locations [2, 6, 7, 11, 14, 17]. Unfortunately, cloaking methods are not robust with respect to composition. For instance, reporting the area is subject to triangulation attacks. Furthermore, they require assumptions about the attacker's side information. For example, dummy locations are only useful if they look equally likely to be the real location from the point of view of the attacker.

A second class of location obfuscation mechanisms involve the generation of controlled noise for Bayesian adversaries. We mention in particular [10] and [15]: The first obtains a perturbation mechanism by crossing paths of individual users, thus rendering the task of tracking individual paths challenging. The second obtains an optimal mechanism (i.e., achieving maximum level of privacy for the user) by solving a linear program in which the constraints are determined by the quality of service and by the user’s profile.

## 1.2 Plan of the Paper

In the next section we present our basic approach to location privacy, based on the notion of geo-indistinguishability. In Sect. 3 we then discuss the problems that raise when we repeatedly use the mechanism along a trace, and when the space is not uniform from the point of view of location hiding, and we illustrate our approach to address these problems. Finally, Sect. 4 presents some future work.

## 2 Geo-Indistinguishability



**Fig. 1.** The prob. density functions of two planar Laplacians, centered on the (real) locations  $(-2, -4)$  and  $(5, 3)$  respectively.

domains, obtained by replacing the Hamming distance, implicit in the definition of differential privacy, with the intended distance – namely the geographical distance in our case. Like differential privacy, geo-indistinguishability is independent from the side knowledge of the adversary and robust with respect to composition of attacks.

We have implemented geo-indistinguishability by adding random noise drawn from a planar Laplace distribution, see Fig. 1. In [1] we have compared this mechanism with the representatives of the other methods proposed in the literature (the cloaking and the linear programming mechanisms), using the privacy metric

Our approach is based on the property of *geo-indistinguishability* [1], which guarantees that the user’s location is protected, within a radius  $r$ , with a level of noise that decreases with  $r$ , at a rate that depends on the desired level of privacy. Intuitively, this means that the real location is highly indistinguishable from the locations that are close, and gradually more distinguishable from those that are far away. This characteristics allows us to obtain a good level of privacy without significant loss of utility.

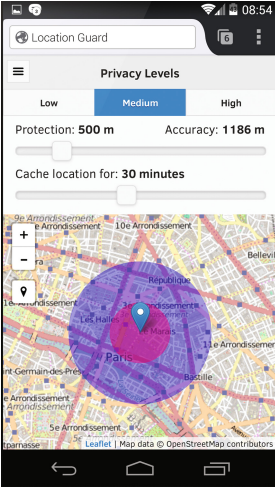
From a technical point of view, geo-indistinguishability is a particular instance of  $d$ -privacy [4], an extension of *differential privacy* [8] to arbitrary metric

proposed in [15]. It turns out that our mechanism offers the best privacy guarantees, for the same utility, among all those which do not depend on the prior knowledge of the adversary. The advantages of the independence from the prior are obvious: first, the mechanism is designed once and for all (i.e. it does not need to be recomputed every time the adversary changes, it works also in simultaneous presence of different adversaries, etc.). Second, and even more important, it is applicable also when we do not know the prior.

Our technique can be used to enhance any application for location-based services with privacy guarantees, and can be implemented on the client side of the application. To this purpose, we are developing a tool, called Location Guard.

## 2.1 Location Guard

Location Guard [<https://github.com/chatziko/location-guard>] is an open source web browser extension based on geo-indistinguishability, that provides location privacy when using the HTML5 geolocation API (Fig. 2).



**Fig. 2.** Privacy level configuration on Android,  $r_u$  in purple and  $r_p$  in pink.

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.

Location Guard has reached considerable popularity since its release in Fall 2014, covering Chrome, Firefox and Opera browsers, and more recently moving to mobile devices with Firefox for Android. As of June 2015 Location Guard counts 9,800 active users in Google Chrome, 29,400 in Mozilla Firefox (including Android) and 5,000 downloads in Opera. Adoption has been mainly through the browser extension stores, as well as through technology blogs covering Location Guard [3, 13]. In June 2015 it was chosen as “Pick of the Month” in Mozilla Add-ons Blog [16].

### 3 Making Geo-Indistinguishability Flexible Over Time and Space.

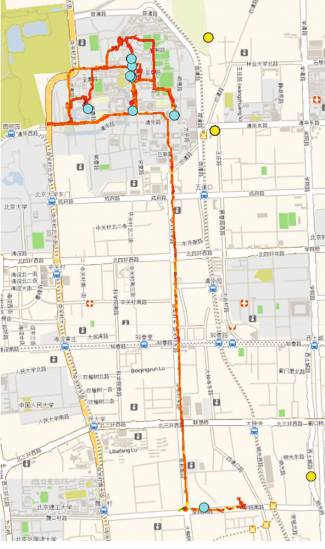
Geo-indistinguishability and its current implementation Location Guard are just a preliminary approach to location privacy, and they present two main limitations. First, when used repeatedly, there is a linear degradation of the user’s privacy that limits the use of the mechanism over time. Second, the level of noise of the Laplacian mechanism has to be fixed in advance independently of the movements of the user, providing the same protection in areas with very different privacy characteristic, like a dense city or a sparse countryside. This limits the flexibility of the mechanism over space.

In this section we present two extensions that we developed to overcome these issues as well as future challenges that we plan to tackle. Many of techniques presented are currently being introduced into Location Guard, in order to extend its range of applications and at the same time provide a realistic experimentation platform to evaluate them.

#### 3.1 Repeated Use Over Time

The main limitation of Location Guard is that, so far, it works well when used sporadically, to protect a single location, for instance when querying an LBS to find some point of interest (restaurants, cinemas, ...) in the vicinity.

We aim at extending the range of applications by handling traces (sequences of location points). This is a very challenging task. Note, in fact, that the naive approach of applying the noise at every step would cause a dramatic privacy degradation, due to the large number of points. Intuitively, in the extreme case when the user never moves (which corresponds to maximum correlation), the reported locations would be centered around the real one, thus revealing it more and more precisely as the number of queries increases. Technically, the independent mechanism applying  $\epsilon$ -geo-indistinguishable noise (where  $\epsilon$  is the privacy parameter) to  $n$  locations can be shown to satisfy  $n\epsilon$ -geo-indistinguishability. This is a typical phenomenon in the framework of differential privacy, and consequently  $n\epsilon$  is thought as a privacy *budget*, consumed by each query. This linear increase makes the mechanism applicable only when the number of queries remains small.



**Fig. 3.** Original trace (red), sampled trace (light blue) and reported trace (yellow) (Color figure online).

while moving around the city. The prediction function used is simply behaving like a cache: It predicts that the user doesn’t move and that the next location will be the same as the last one. This prediction function has the advantages of being trivial to implement, independent of the user profile and proved to be very effective in our evaluation.

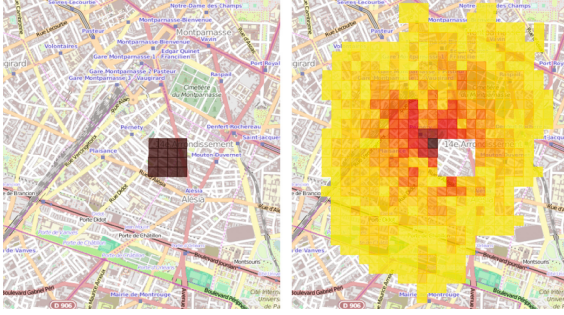
*Example of Sanitized Trace.* Fig. 3 displays one of Geolife trajectories sanitized with fixed utility. The original trace, in red, starts south with low speed, moves north on a high speed road and then turns around Tsinghua University for some time. In order to model a user’s sporadic behavior we sample the trace obtaining the 9 light blue dots, which are locations where the user queries the LBS. Finally in yellow we have the reported trace, sanitized by the predictive mechanism, with only 3 locations. The first used once for the point at the bottom, the second 7 times for the one in the middle and the third twice for point in the top. In this example the mechanism needed to sanitize with noise only 3 locations, using them as prediction for the other 6.

### 3.2 Highly Recurrent Locations

Even with the budget savings of the predictive mechanism, the user’s privacy is bound to be breached in the long run in those locations that are *highly recurrent*, such as home and work. We propose a simple construction to model “geographic fences”: Areas around highly recurrent locations where the mechanism reports uniformly, effectively stopping the privacy erosion. On one side the user has to

In [5] we explore a *trace obfuscation* mechanism with a smaller *budget consumption rate* than the one produced by applying independent noise. We show that correlation in the trace can be in fact exploited through a *prediction function* that tries to guess the new location based on the previously reported locations. Predicted points are safe to report directly (the adversary would have guessed them in any case) and thus have a smaller footprint on the privacy budget, because they reduce the need of applying the noise at every step. However the inclusion of the prediction function in a privacy mechanism has to be private itself, leading to additional costs for the privacy budget of the user. If there is considerable correlation in the input trace, our carefully designed *budget managers* handle this balance of costs, producing a more efficient *predictive mechanism*.

The mechanism is evaluated using the Geolife and T-Drive datasets, containing traces of thousands of users in the Beijing area. The users are modeled as accessing a location-based service



**Fig. 4.** Probability distribution of reported location inside and outside the fence. Darker colors indicate more likely values (Color figure online).

release publicly the position of her fences but on the other the budget cost when reporting from inside them is zero, leading to a practical solution that can be used in combination with the predictive mechanism.

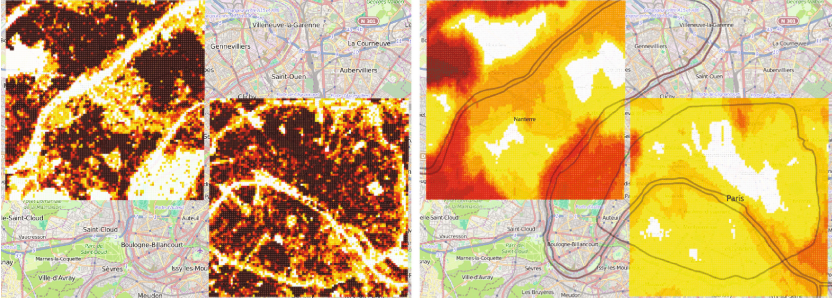
In Fig. 4 we can see an example of fence introduced in an elastic metric. On the left we have the distribution of reported locations inside the fence, that is perfectly uniform, covering a few blocks and proving an adequate level of privacy while costing zero on the budget. On the right we can see the distribution of reported locations of a point right outside, the fence is clearly visible and the mechanism reports right around it.

### 3.3 Flexible Behavior Over Space

Another shortcoming of standard geo-indistinguishability is that the privacy level has to be fixed independently of the user location. 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 more depth in [12], where we propose an *elastic mechanism* that adapts the level of noise to the semantic characteristics of each location, such as population and presence of POIs. We perform an extensive evaluation of our technique by building an elastic mechanism for Paris’ wide metropolitan area, using semantic information from the OpenStreetMap database.

The resulting privacy *mass* of each location is shown in Fig. 5a, where white color indicates a small mass while yellow, red and black indicate increasingly greater mass. The figure is just a small extract of the whole grid depicting the two smaller areas used in the evaluation: central Paris and the nearby suburb of Nanterre. Note that the colors alone depict a fairly clear picture of the city: in white we can see the river traversing horizontally, the main ring-road and several spots mark parks and gardens. In yellow colors we find low density areas as well as roads and railways while red colors are present in residential areas. Finally dark colors indicate densely populated areas with presence of POIs.





(a) Privacy mass of each location      (b) Expected error at each location

**Fig. 5.** Paris' center (right) and the nearby suburb of Nanterre (left)

Figure 5b shows our utility per location, computed as the expected distance between the real and the reported location. Compared to Fig. 5a it is clear that areas with higher privacy mass result to less noise. Populated areas present a good and uniform error that starts to increase on the river and ring-road. On the other hand, the large low-density areas, especially in the Nanterre suburb, have a higher error because they need to report over larger areas to reach the needed amount of privacy.

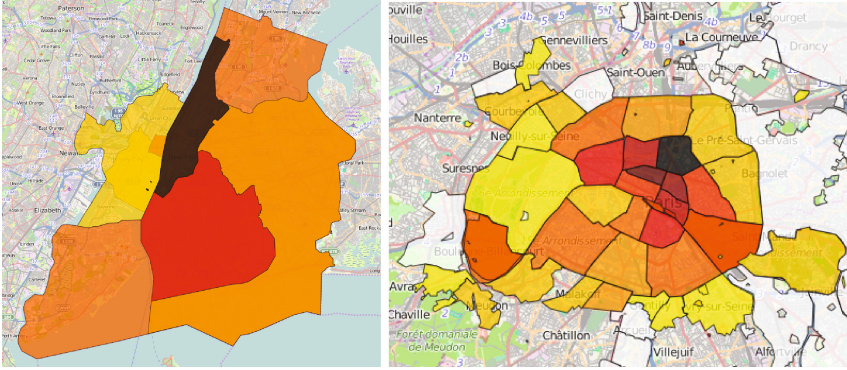
We compare the resulting mechanism against the Planar Laplace mechanism satisfying standard geo-indistinguishability, using two real-world datasets from the Gowalla and Brightkite location-based social networks. The results show that the elastic mechanism adapts well to the semantics of each area, adjusting the noise as we move outside the city center, hence offering better overall privacy.

### 3.4 A Tiled Mechanism

The extreme flexibility of the elastic 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.

For this reason we propose a lighter version of the *elastic mechanism*, that requires no pre-computation of the metric, and is thus suitable for lower end devices and for an easier inclusion in existing systems. Of course this *tiled mechanism* provides less flexibility: Instead of adapting the noise differently in locations tens of meters apart, it can only adapt to large areas of a city, covering tens of square kilometers. These areas, that we call tiles, are small enough to distinguish a park from a residential area, but still easily computable. In order to build the set of tiles, we query two online geographical services, **overpass-turbo** and **dbpedia** to obtain a set of polygons together with a quantitative description of the amount of privacy they provide. 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 built, we would have a mapping from tiles to their privacy mass, and we would





**Fig. 6.** Polygons computed for New York and Paris

use it to define a function  $\ell$  that, for each location, finds the containing polygon and returns a privacy level adapted to the privacy mass provided by the tile. Examples of the kind of maps that we aim at obtaining with this method are shown in Fig. 6.

The mechanism described above, despite achieving the flexible behavior we needed, would 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 would need to make  $\ell$  itself differentially private. A simple way to do it could be to first sanitize the current location with a fixed privacy level and then feed it to  $\ell$ . 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.

## 4 Future Work

Regarding the geographic fences we are currently evaluating how to automatically configure their position and size. The user input would be the best option, however they could also be inferred and suggested automatically. In [9] the authors developed an attack to identify POI of a specific user, from a set of mobility traces. A similar technique could be employed on the user's phone, over a training period, to collect and analyze her movements for a few days. The mechanism would then automatically detect recurrent locations and suggest the user to fence them, possibly detecting more than just home/work locations.

With the use of geolocated queries, such as those used to extract privacy mass of the elastic mechanism, we could determine the size of the fence so to include a reasonable amount of buildings for home and other POIs for work.

Concerning the elastic mechanism in some cases we might want to tailor our mechanism to a specific group of users, to increase the performance in terms of both privacy and utility. In this case, given a *prior* probability distribution over

the grid of locations, we can use it to influence the privacy mass of each cell. For instance, if we know that our users never cross some locations or certain kind of POIs, we can reduce their privacy mass.

Moreover, we are interested in queries that reward variety other than richness e.g. a location with 50 restaurants should be considered less private than one with 25 restaurants and 25 shops.

Finally, different grids could be computed for certain periods of the day or of the year. For instance, our user could use the map described above during the day, feeling private in a road with shops, but in the evening only a subset of the tags should be used as many activities are closed, making a road with many restaurants a much better choice. The same could be applied to seasons, imagine for example how snow affects human activities in many regions.

Additionally we are also actively working on the tiled mechanism in order to provide both a formal proof of privacy as well as an efficient implementation to include in Location Guard.

## References

1. Andrés, M.E., Bordenabe, N.E., Chatzikokolakis, K., Palamidessi, C.: Geo-indistinguishability: differential privacy for location-based systems. In: *Proceedings of CCS*, pp. 901–914. ACM (2013)
2. Bamba, B., Liu, L., Pesti, P., Wang, T.: Supporting anonymous location queries in mobile environments with privacygrid. In: *Proceedings of WWW*, pp. 237–246. ACM (2008)
3. Brinkmann, M.: Change your location in firefox using location guard. Ghacks.net (2014)
4. Chatzikokolakis, K., Andrés, M.E., Bordenabe, N.E., Palamidessi, C.: Broadening the scope of differential privacy using metrics. In: De Cristofaro, E., Wright, M. (eds.) *PETS 2013*. LNCS, vol. 7981, pp. 82–102. Springer, Heidelberg (2013)
5. Chatzikokolakis, K., Palamidessi, C., Stronati, M.: A predictive differentially-private mechanism for mobility traces. In: De Cristofaro, E., Murdoch, S.J. (eds.) *PETS 2014*. LNCS, vol. 8555, pp. 21–41. Springer, Heidelberg (2014)
6. Cheng, R., Zhang, Y., Bertino, E., Prabhakar, S.: Preserving user location privacy in mobile data management infrastructures. In: Danezis, G., Golle, P. (eds.) *PET 2006*. LNCS, vol. 4258, pp. 393–412. Springer, Heidelberg (2006)
7. Duckham, M., Kulik, L.: A formal model of obfuscation and negotiation for location privacy. In: Gellersen, H.-W., Want, R., Schmidt, A. (eds.) *PERVASIVE 2005*. LNCS, vol. 3468, pp. 152–170. Springer, Heidelberg (2005)
8. Dwork, C.: Differential privacy. In: Bugliesi, M., Preneel, B., Sassone, V., Wegener, I. (eds.) *ICALP 2006*. LNCS, vol. 4052, pp. 1–12. Springer, Heidelberg (2006)
9. Gambs, S., Killijian, M.O., del Prado Cortez, M.N.: Show me how you move and i will tell you who you are. *Trans. Data Priv.* **4**(2), 103–126 (2011)
10. Hoh, B., Gruteser, M.: Protecting location privacy through path confusion. In: *Proceedings of SecureComm*, pp. 194–205. IEEE (2005)
11. Kido, H., Yanagisawa, Y., Satoh, T.: Protection of location privacy using dummies for location-based services. In: *Proceedings of ICDE Workshops*, p. 1248 (2005)
12. Stronati, M., Chatzikokolakis, K., Palamidessi, C.: Constructing elastic distinguishability metrics for location privacy. In: *Proceedings of PETS* (2015). To appear

13. Korben, D.: Géolocalisation - restez maître de votre situation (2015). <http://korben.info/geolocalisation-restez-maitre-de-votre-situation.html>
14. Shankar, P., Ganapathy, V., Iftode, L.: Privately querying location-based services with SybilQuery. In: Proceedings of UbiComp, pp. 31–40. ACM (2009)
15. Shokri, R., Theodorakopoulos, G., Troncoso, C., Hubaux, J.P., Le Boudec, J.Y.: Protecting location privacy: optimal strategy against localization attacks. In: Proceedings of CCS, pp. 617–627. ACM (2012)
16. Tsay, A.: Mozilla add-ons blog, June 2015. <https://blog.mozilla.org/addons/2015/06/01/june-2015-featured-add-ons/>
17. Xue, M., Kalnis, P., Pung, H.K.: Location diversity: enhanced privacy protection in location based services. In: Choudhury, T., Quigley, A., Strang, T., Suginuma, K. (eds.) LoCA 2009. LNCS, vol. 5561, pp. 70–87. Springer, Heidelberg (2009)

Theoretical Aspects of Computing - ICTAC 2015  
12th International Colloquium, Cali, Colombia, October  
29-31, 2015, Proceedings  
Leucker, M.; Rueda, C.; Valencia, F.D. (Eds.)  
2015, XXIV, 620 p. 142 illus. in color., Softcover  
ISBN: 978-3-319-25149-3