

Foundations of Privacy

Lecture 4 - Part II

Motivation

Can differential privacy be adapted to different **privacy requirements**?

Can we use differential privacy on secrets that are **not databases**?

Outline

- ▶ **Generalization of differential privacy**
- ▶ Privacy in the context of statistical databases
- ▶ Privacy in location-based systems

Differential Privacy, adjacent databases

- ▶ **Adjacency**: $x \sim_h x'$ iff they differ in exactly one individual

$$x = \langle 32, 41, 27 \rangle$$

$$x' = \langle 32, 52, 27 \rangle$$

- ▶ $K : \mathcal{X} \rightarrow \mathcal{P}(\mathcal{Z})$ satisfies ϵ -differential privacy iff

$$K(x)(Z) \leq e^\epsilon K(x')(Z) \quad \forall x \sim_h x'$$

- ▶ ϵ : **distinguishability level** between adjacent databases

Differential Privacy, any databases

- ▶ Hamming distance $d_h(x, x')$: # of elements in which x, x' differ

$$x = \langle 32, 41, 27 \rangle$$

$$x' = \langle 21, 52, 27 \rangle$$

$$d_h(x, x') = 2$$

Differential Privacy, any databases

- ▶ Hamming distance $d_h(x, x')$: # of elements in which x, x' differ

$$x = \langle 32, 41, 27 \rangle$$

$$x' = \langle 21, 52, 27 \rangle$$

$$d_h(x, x') = 2$$

- ▶ Differential privacy can be equivalently defined as follows:

$$K(x)(Z) \leq e^{\epsilon d_h(x, x')} K(x')(Z) \quad \forall x, x'$$

Differential Privacy, any databases

- ▶ **Hamming distance** $d_h(x, x')$: # of elements in which x, x' differ

$$x = \langle 32, 41, 27 \rangle$$

$$x' = \langle 21, 52, 27 \rangle$$

$$d_h(x, x') = 2$$

- ▶ Differential privacy can be equivalently defined as follows:

$$K(x)(Z) \leq e^{\epsilon d_h(x, x')} K(x')(Z) \quad \forall x, x'$$

- ▶ $\epsilon d_h(x, x')$: distinguishability level between **any** databases

Differential Privacy, any databases

- ▶ **Hamming distance** $d_h(x, x')$: # of elements in which x, x' differ

$$x = \langle 32, 41, 27 \rangle$$

$$x' = \langle 21, 52, 27 \rangle$$

$$d_h(x, x') = 2$$

- ▶ Differential privacy can be equivalently defined as follows:

$$K(x)(Z) \leq e^{\epsilon d_h(x, x')} K(x')(Z) \quad \forall x, x'$$

- ▶ $\epsilon d_h(x, x')$: distinguishability level between **any** databases
- ▶ the **less distinguishable** two databases are, the **more similar** the outcome should be

Differential Privacy, generalization

- ▶ Arbitrary domain of secrets \mathcal{X}

Differential Privacy, generalization

- ▶ Arbitrary domain of secrets \mathcal{X}
- ▶ $\epsilon(x, x')$: distinguishability level between x, x'

Differential Privacy, generalization

- ▶ Arbitrary domain of secrets \mathcal{X}
- ▶ $\epsilon(x, x')$: distinguishability level between x, x'
- ▶ Expected properties:
 - ▶ $\epsilon(x, x) = 0$
 - ▶ $\epsilon(x, x') = \epsilon(x', x)$
 - ▶ $\left. \begin{array}{l} \epsilon(x_1, x_2) \leq b \\ \epsilon(x_3, x_2) \leq b \end{array} \right\} \Rightarrow \epsilon(x_1, x_3) \leq f(b)$

Differential Privacy, generalization

- ▶ Arbitrary domain of secrets \mathcal{X}
- ▶ $\epsilon(x, x')$: distinguishability level between x, x'
- ▶ Expected properties:
 - ▶ $\epsilon(x, x) = 0$
 - ▶ $\epsilon(x, x') = \epsilon(x', x)$
 - ▶ $\left. \begin{array}{l} \epsilon(x_1, x_2) \leq b \\ \epsilon(x_3, x_2) \leq b \end{array} \right\} \Rightarrow \epsilon(x_1, x_3) \leq f(b)$
- ▶ We take $\epsilon(x, x')$ to be a **metric**, denoted d_x

$$d_x(x_1, x_3) \leq d_x(x_1, x_2) + d_x(x_3, x_2)$$

Differential Privacy, generalization

d_x -privacy

$$K(x)(Z) \leq e^{d_x(x,x')} K(x')(Z) \quad \forall x, x'$$

- ▶ the **less distinguishable** two secrets are, the **more similar** the outcome should be
- ▶ There is no ϵ , but we can just rescale the metric in order to obtain the desired level of privacy: $d_x = \epsilon d_{x'}$
- ▶ ϵ -differential privacy = ϵd_h -privacy

Differential Privacy, generalization

d_x -privacy

$$K(x)(Z) \leq e^{d_x(x,x')} K(x')(Z) \quad \forall x, x'$$

This notion of privacy protects the **accuracy** of the data

- ▶ **Foundations**

- ▶ Compositionality
- ▶ Implementation: Laplacian
- ▶ Optimality results

- ▶ **Applications**

- ▶ Statistical databases - (normalized) Manhattan distance
- ▶ Location privacy - Geographical distance
- ▶ In general, every domain equipped with a metric

Compositionality

If K, K' are d_x and $d_{x'}$ differentially private, then the composition of the two mechanisms, (K, K') , is $d_x + d_{x'}$ differentially private

Answering queries

- ▶ Query $f : \mathcal{X} \rightarrow \mathcal{Y}$
- ▶ f is Δ -sensitive wrt d_x, d_y iff:

$$\Delta = \max_{x, x'} \frac{d_y(f(x), f(x'))}{d_x(x, x')}$$

- ▶ If $H : \mathcal{Y} \rightarrow \mathcal{P}(\mathcal{Z})$ satisfies d_y -privacy then $H \circ f$ satisfies Δd_x -privacy
- ▶ H can be implemented in the usual way as **Laplacian noise**:

$$H(y)(z) = c \cdot e^{\frac{-d_y(z, y)}{\Delta} \epsilon}$$

We can easily prove that H satisfies $\frac{d_y}{\Delta} \epsilon$ -privacy, and consequently $H \circ f$ satisfies $d_x \epsilon$ -privacy

Outline

- ▶ Generalization of differential privacy
- ▶ **Privacy in the context of statistical databases**
- ▶ Privacy in location-based systems

The normalized Manhattan metric

- ▶ The Hamming distance is independent from the actual values

$$x_1 = \langle 32, 0, 27 \rangle$$

$$x_2 = \langle 32, 0.01, 27 \rangle$$

$$x_3 = \langle 32, 10^6, 27 \rangle$$

$$d_h(x_1, x_2) = d_h(x_1, x_3) = 1$$

The normalized Manhattan metric

- ▶ The Hamming distance is independent from the actual values

$$x_1 = \langle 32, 0, 27 \rangle$$

$$x_2 = \langle 32, 0.01, 27 \rangle$$

$$x_3 = \langle 32, 10^6, 27 \rangle \quad d_h(x_1, x_2) = d_h(x_1, x_3) = 1$$

- ▶ the disting. level between x_1, x_2 and x_2, x_3 is the same

The normalized Manhattan metric

- ▶ The Hamming distance is independent from the actual values

$$x_1 = \langle 32, 0, 27 \rangle$$

$$x_2 = \langle 32, 0.01, 27 \rangle$$

$$x_3 = \langle 32, 10^6, 27 \rangle \quad d_h(x_1, x_2) = d_h(x_1, x_3) = 1$$

- ▶ the disting. level between x_1, x_2 and x_2, x_3 is the same
- ▶ Many queries are insensitive to minor changes in values

The normalized Manhattan metric

- ▶ The Hamming distance is independent from the actual values

$$x_1 = \langle 32, 0, 27 \rangle$$

$$x_2 = \langle 32, 0.01, 27 \rangle$$

$$x_3 = \langle 32, 10^6, 27 \rangle \quad d_h(x_1, x_2) = d_h(x_1, x_3) = 1$$

- ▶ the disting. level between x_1, x_2 and x_2, x_3 is the same
- ▶ Many queries are insensitive to minor changes in values
- ▶ If ϵ is “weak”, we might require higher protection for x_1, x_2

The normalized Manhattan metric

- ▶ Manhattan metric:

$$d_1(x, x') = \sum_{i=1}^n d_v(x[i], x'[i])$$

- ▶ Normalized Manhattan metric:

$$\tilde{d}_1(x, x') = \frac{d_1(x, x')}{d_v(\mathcal{V})}$$

where $d_v(\mathcal{V})$ is the maximum distance among the values

- ▶ Stronger than Hamming: $\tilde{d}_1(x, x') \leq d_h(x, x')$

$$x_1 = \langle 32, 0, 27 \rangle$$

$$x_2 = \langle 32, 0.01, 27 \rangle$$

$$x_3 = \langle 32, 10^6, 27 \rangle$$

$$\tilde{d}_1(x_1, x_2) = 10^{-8}$$

$$\tilde{d}_1(x_1, x_3) = 1$$

Advantages of the normalized Manhattan metric

Sensitivity:

- ▶ For a family of queries (sum, average, percentile, . . .), the sensitivity wrt \tilde{d}_1 , $d_{\mathbb{R}}$ and d_h , $d_{\mathbb{R}}$ coincide
- ▶ In general, \tilde{d}_1 is smaller than d_h
- ▶ hence we get **stronger privacy** with the same noise

Optimality:

- ▶ If the set of values is discrete, then sum, average and percentile queries induce a graph structure which is **a straight line**
- ▶ As a consequence, the Geometric mechanism is **universally optimal** for sum, average and percentile queries wrt \tilde{d}_1
- ▶ In contrast, we saw that only counting queries have universally optimal mechanisms wrt d_h

The Manhattan metric

- ▶ We can use the Manhattan metric without normalization:

$$d_1(x, x') = \sum_{i=1}^n d_v(x[i], x'[i])$$

- ▶ d_1 can be much higher than Hamming, but Δ will be proportionally smaller than the usual sensitivity, so the protection, with respect to the introduced noise, is comparable.

Example:

$$x_1 = \langle 32, 0, 27 \rangle$$

$$x_2 = \langle 32, 0.01, 27 \rangle$$

$$x_3 = \langle 32, 10^6, 27 \rangle$$

$$\tilde{d}_1(x_1, x_2) = 10^{-2}$$

$$\tilde{d}_1(x_1, x_3) = 10^6$$

The Manhattan metric

- ▶ The Manhattan metric be useful when we need to prevent the attacker from getting very precise data (for instance because they can be used to **identify** an individual),
- ▶ Trade-off between **privacy** and **utility**
- ▶ **Optimality results** similar to \tilde{d}_1

Outline

- ▶ Generalization of differential privacy
- ▶ Privacy in the context of statistical databases
- ▶ **Privacy in location-based systems**

Motivation

Geographical information is becoming essential for a variety of services: LBS, advertising, social networks, data mining, . . .

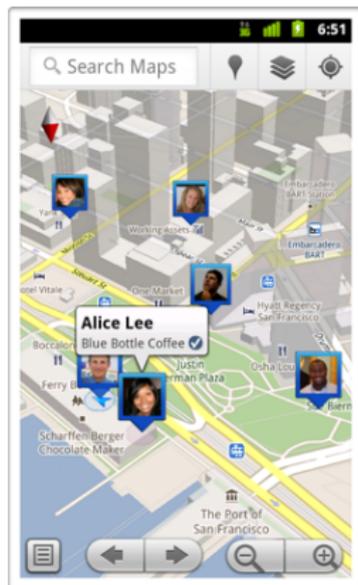


Privacy: location data are often **sensitive** and need protection

Location-Based Systems

A **location-based system** is a system that uses geographical information in order to provide a service.

- ▶ Retrieval of Points of Interest (POIs).
- ▶ Mapping Applications.
- ▶ Deals and discounts applications.
- ▶ Location-Aware Social Networks.

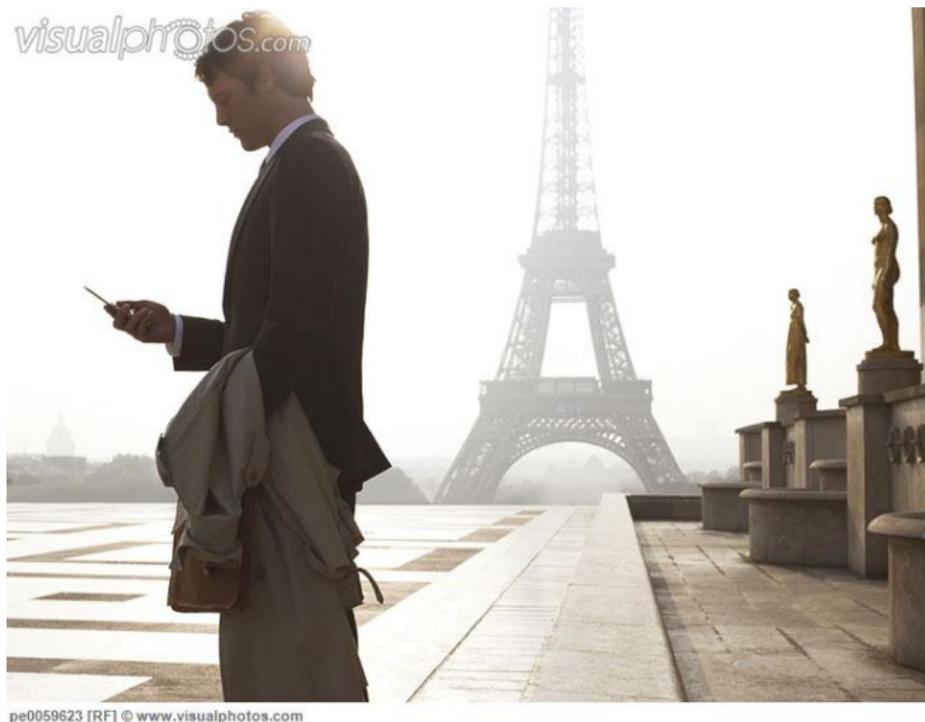


Location-Based Systems

- ▶ **Location information is sensitive.** (it can be linked to home, work, religion, political views, etc).
- ▶ Ideally: we want to **hide our true location.**
- ▶ Reality: we need to **disclose some information.**



Motivating example



Locate a restaurant close to my location

Motivating example

Goal:

- ▶ Hide the user's **location** (not identity) from the **service provider**
- ▶ **Formal** privacy guarantee

Constraints:

- ▶ Implementable in real-time on a smartphone
- ▶ No trusted party
- ▶ Optimally: no peer-to-peer communication

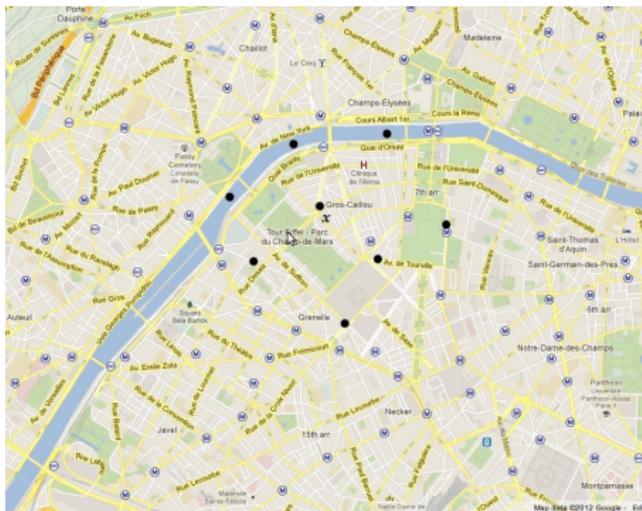


Existing privacy notions

k -anonymity (or l -diversity)

Hide the user's location among k points

- ▶ Include $k - 1$ randomly generated points in the query
- ▶ Use a cloaking region including k points of interest



Problem: depends on the attacker's side information

Existing privacy notions

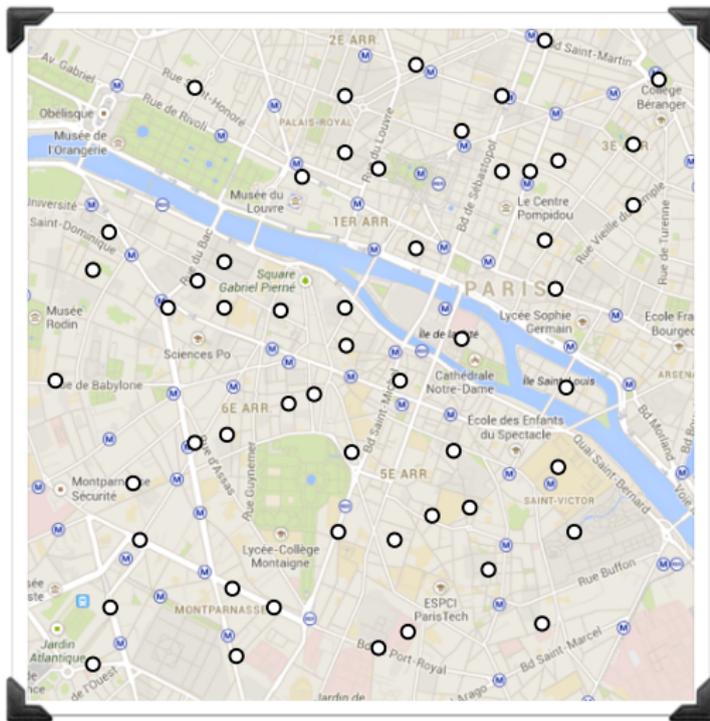
Differential Privacy

Changes in a **single user's** value should have **negligible effect** on the reported value

- ▶ Useful for publishing **aggregate** information about a large number of users
- ▶ Has been used in the context of geo-location
- ▶ Inadequate for our motivating example

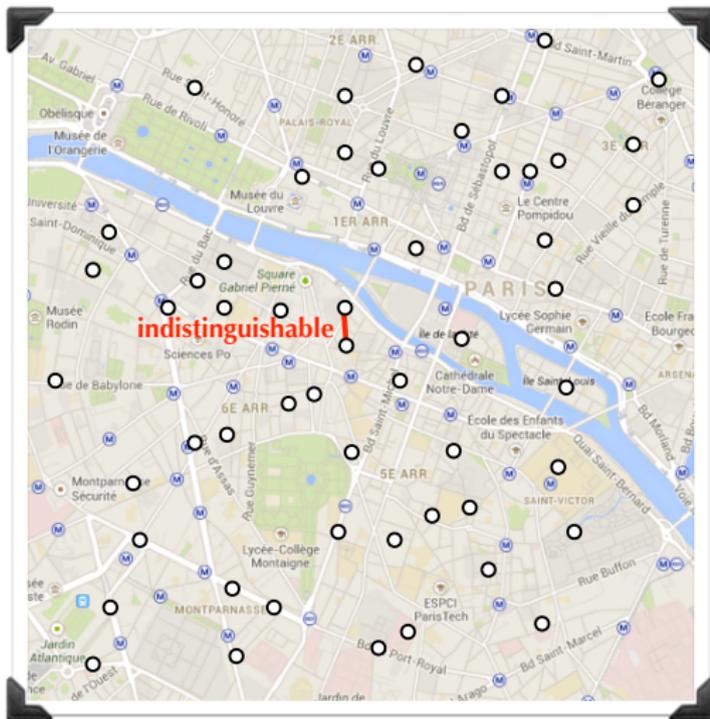
Towards a Definition

- ▶ Secrets are **locations**.
- ▶ Attacker's goal: **distinguish** location x from x' .
- ▶ The closer two locations are, the more indistinguishable they should be.



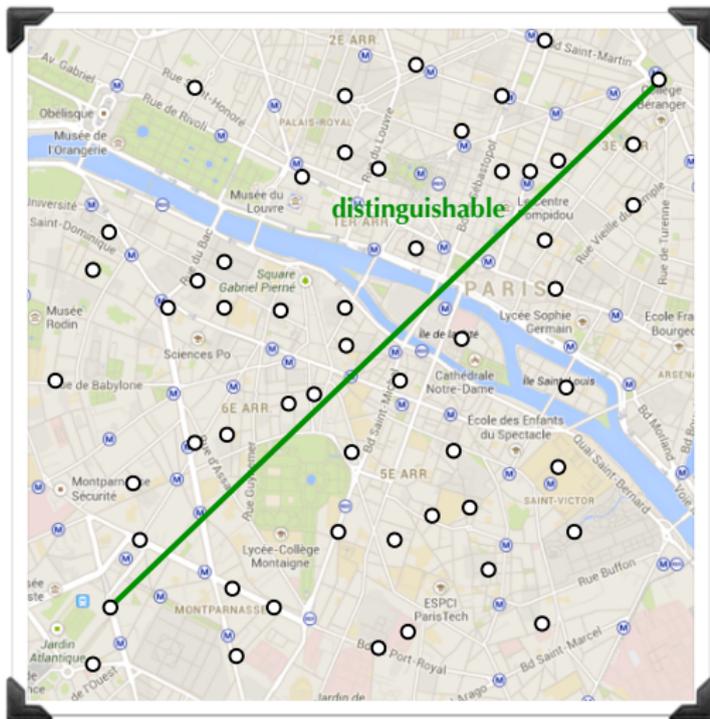
Towards a Definition

- ▶ Secrets are **locations**.
- ▶ Attacker's goal: **distinguish** location x from x' .
- ▶ The closer two locations are, the more indistinguishable they should be.



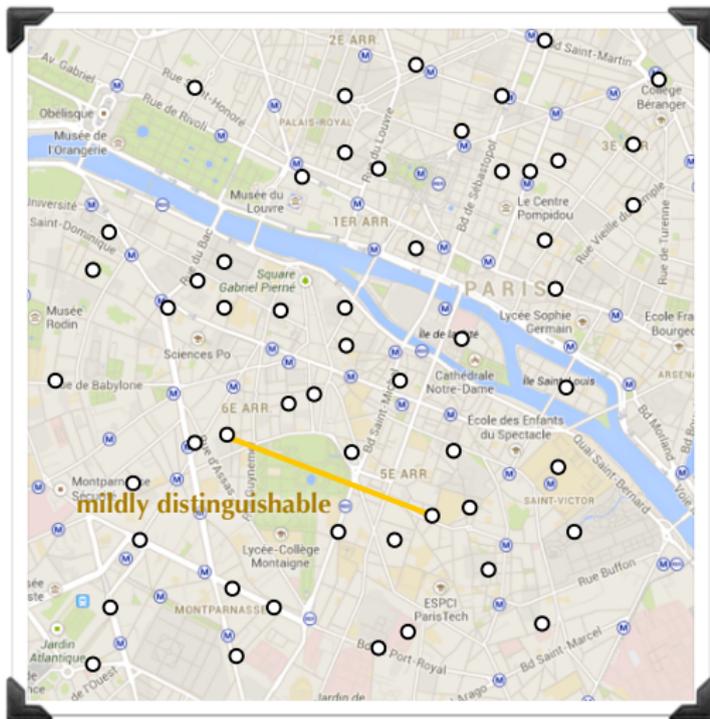
Towards a Definition

- ▶ Secrets are **locations**.
- ▶ Attacker's goal: **distinguish** location x from x' .
- ▶ The closer two locations are, the more indistinguishable they should be.



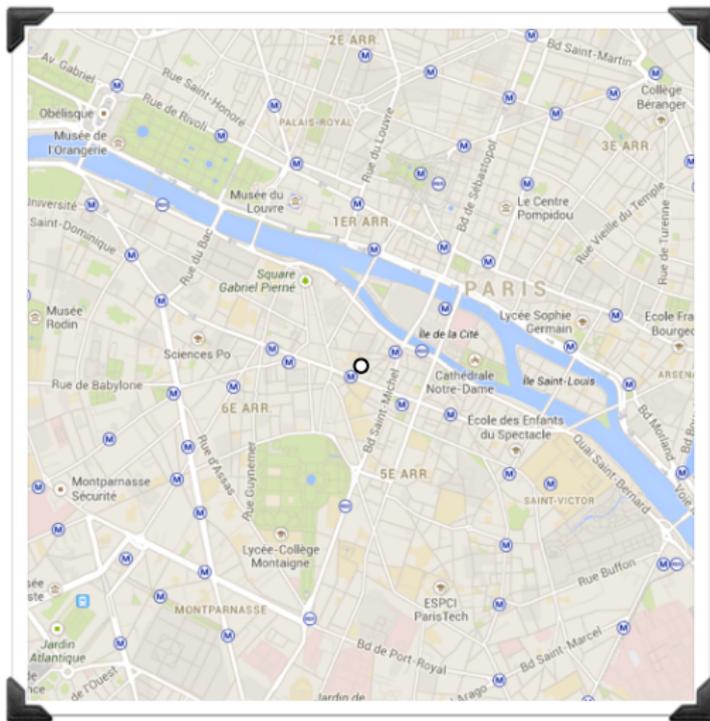
Towards a Definition

- ▶ Secrets are **locations**.
- ▶ Attacker's goal: **distinguish** location x from x' .
- ▶ The closer two locations are, the more indistinguishable they should be.



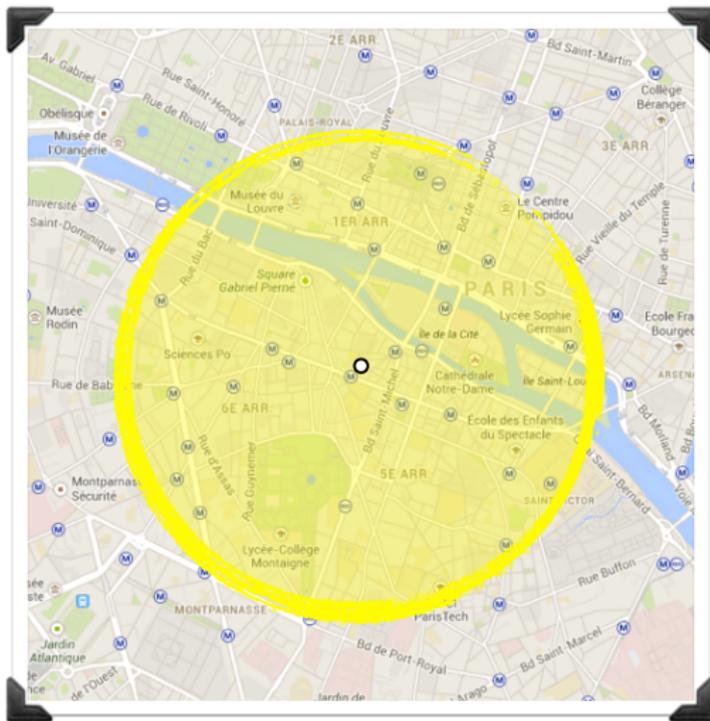
Towards a Definition

- ▶ Secrets are **locations**.
- ▶ Attacker's goal: **distinguish** location x from x' .
- ▶ The closer two locations are, the more indistinguishable they should be.



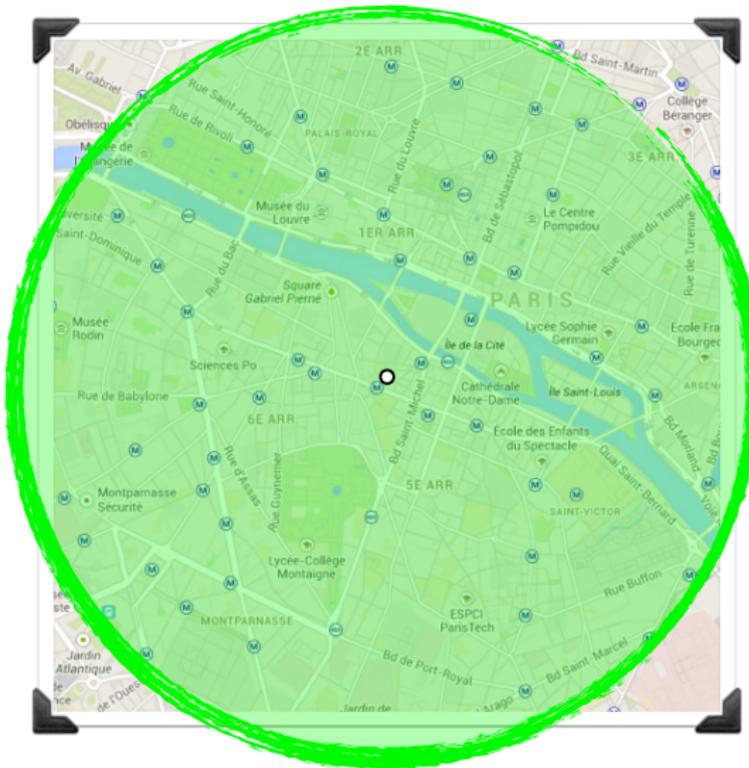
Towards a Definition

- ▶ Secrets are **locations**.
- ▶ Attacker's goal: **distinguish** location x from x' .
- ▶ The closer two locations are, the more indistinguishable they should be.



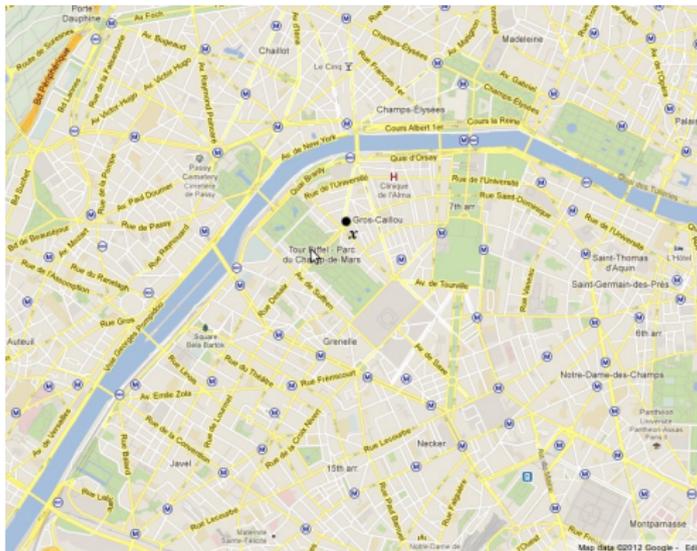
Towards a Definition

- ▶ Secrets are **locations**.
- ▶ Attacker's goal: **distinguish** location x from x' .
- ▶ The closer two locations are, the more indistinguishable they should be.



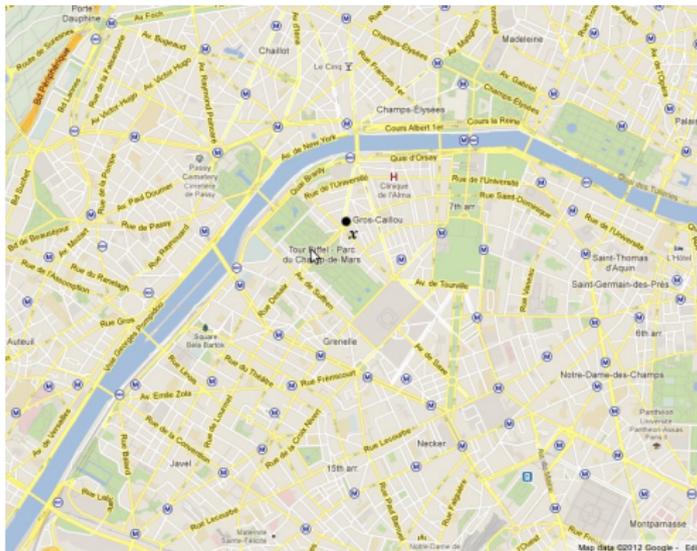
In search for a new definition

- ▶ What kind of privacy does the user **expect** to have?



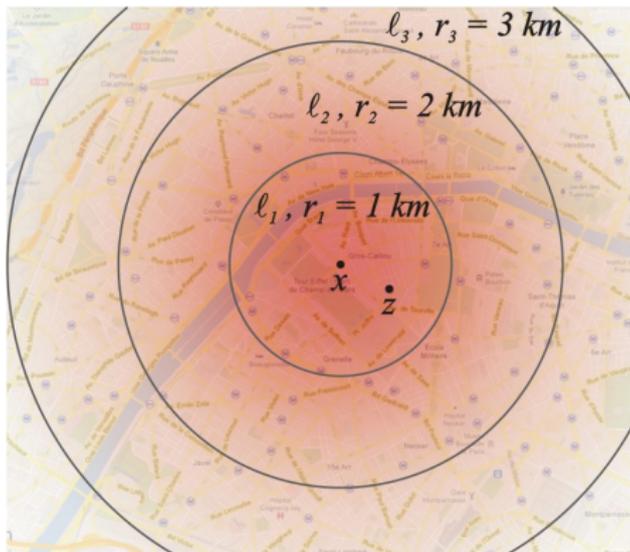
In search for a new definition

- ▶ What kind of privacy does the user **expect** to have?
- ▶ Privacy depends on the **accuracy** of detecting x



In search for a new definition

- ▶ What kind of privacy does the user **expect** to have?
- ▶ Privacy depends on the **accuracy** of detecting x
- ▶ A different **privacy level** / for each **radius** r

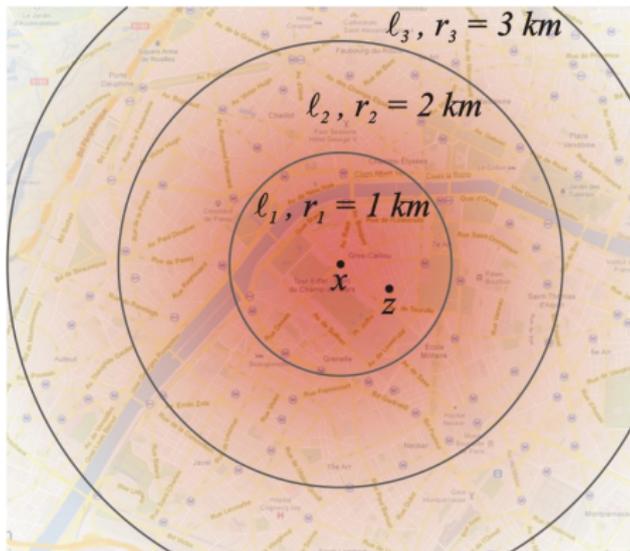


In search for a new definition

ϵ -geo-indistinguishability

Require privacy for **any radius r** with a **proportional level**

$$l(r) = \epsilon \cdot r$$

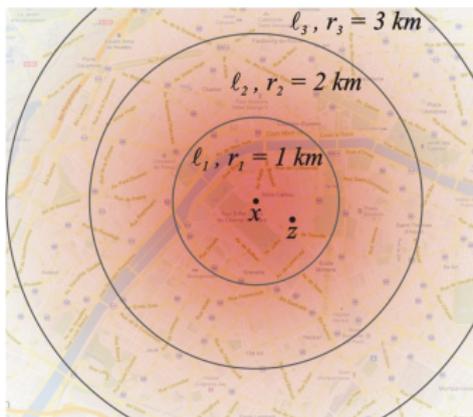


First approach for defining this notion

Intuitively we would like to require:

$$\frac{P(x|z)}{P(x'|z)} \leq e^{\epsilon r} \quad \forall r \forall x, x' : d_2(x, x') \leq r$$

but this might fail because of the **prior knowledge** $P(x)$

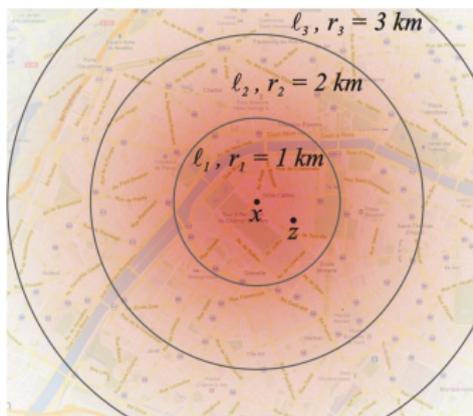


First approach for defining this notion

So we have to take it into account:

$$\frac{P(x|z)}{P(x'|z)} \leq e^{\epsilon r} \frac{P(x)}{P(x')} \quad \forall r \forall x, x' : d_2(x, x') \leq r$$

are require this to hold for any prior $P(x)$

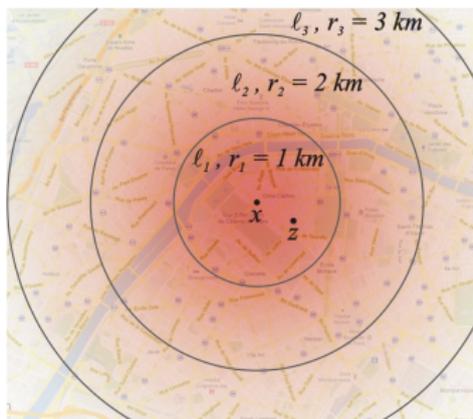


Second approach for defining this notion

Ideally we'd like the **attacker's knowledge** to be **unaffected by z** :

$$\frac{P(x|z)}{P(x)} \leq e^{\epsilon r} \quad \forall r, x$$

but z does provide information (i.e. that the user is in Paris)

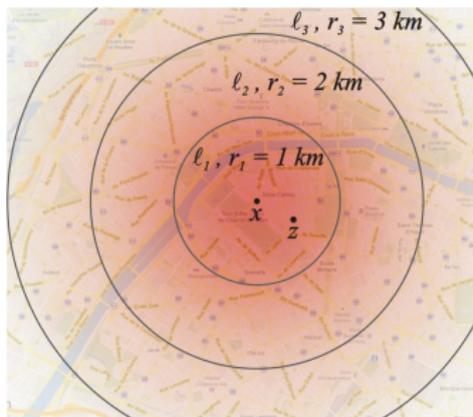


Second approach for defining this notion

So we restrict the increase in knowledge **within the radius r** :

$$\frac{P(x|z, B_r(x))}{P(x|B_r(x))} \leq e^{\epsilon r} \quad \forall r, x$$

again, this should hold for any prior $P(x)$

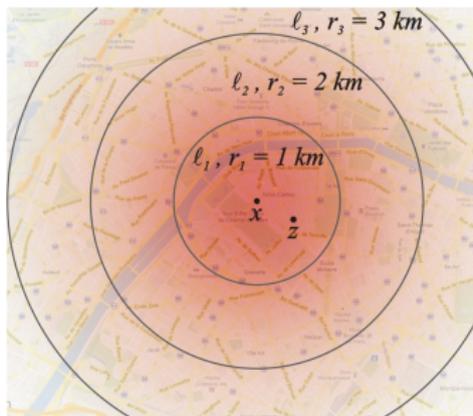


Third approach for defining this notion

Nearby points should produce similar observations:

$$\frac{K(x)(z)}{K(x')(z)} \leq e^{\epsilon r} \quad \forall r \forall x, x' : d_2(x, x') \leq r$$

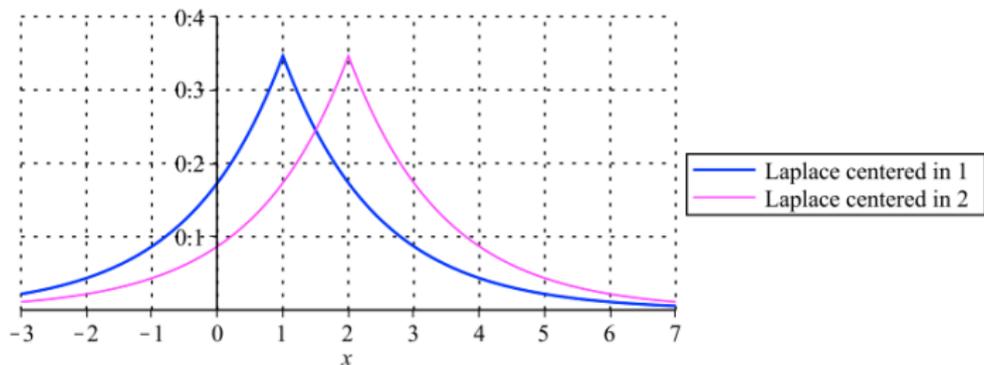
which is the same as ϵd_2 -privacy.



All three formulations are equivalent

A mechanism for geo-indistinguishability

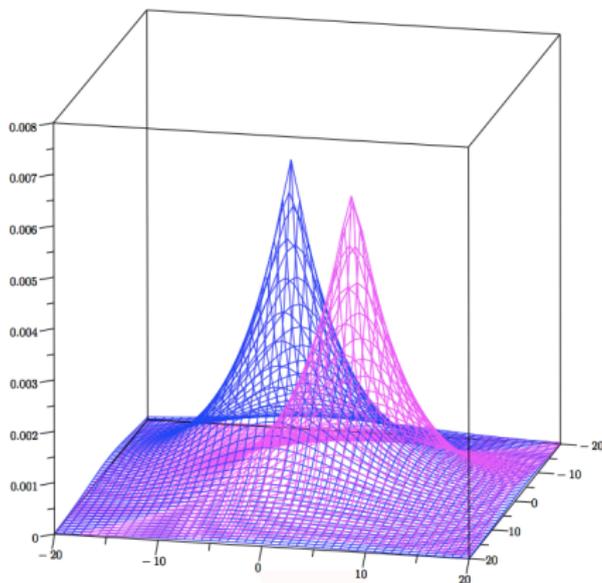
The case of one dimension:



$$\text{pdf: } \frac{\epsilon}{2} e^{-\epsilon|z-x|}$$

A mechanism for geo-indistinguishability

Similarly in two dimensions:

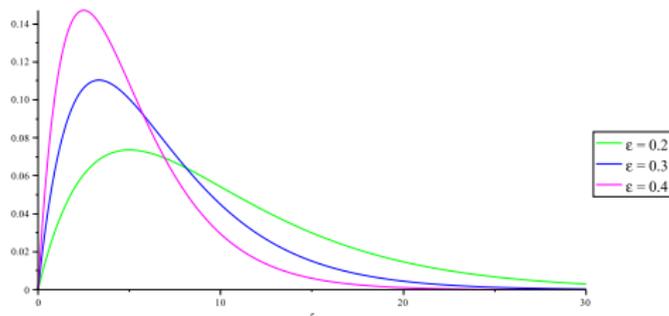


$$\text{pdf: } \frac{\epsilon^2}{2\pi} e^{-\epsilon d_2(\vec{x}, \vec{z})}$$

A mechanism for geo-indistinguishability

Drawing from this distribution:

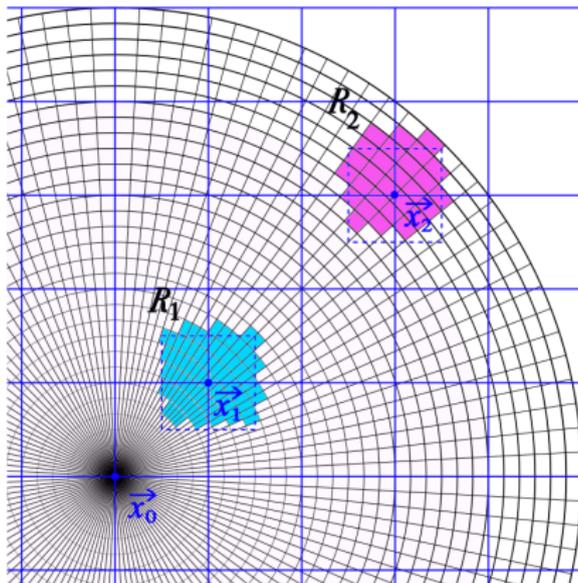
- ▶ use polar coordinates
- ▶ draw an angle θ uniformly
- ▶ draw a radius r from a gamma distribution



$$\text{pdf: } \epsilon^2 r e^{-\epsilon r}$$

A mechanism for geo-indistinguishability

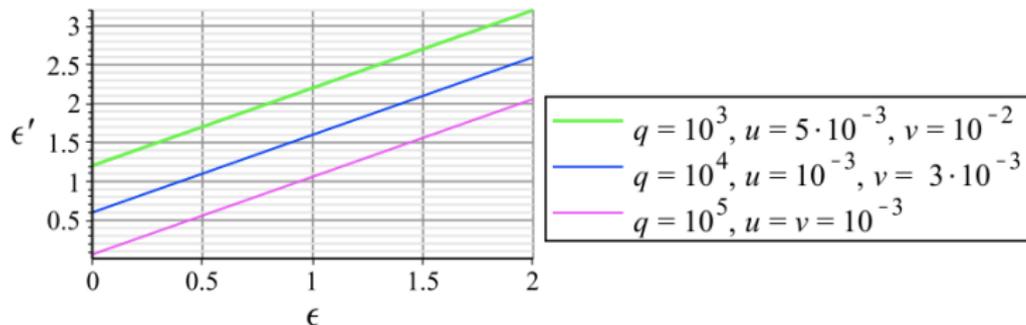
- ▶ In practice locations are discretized
- ▶ (discretely) draw r, θ , map to the **closest point** on the grid
- ▶ Points correspond to **differently shaped areas**, leading to a violation of geo-indistinguishability



A mechanism for geo-indistinguishability

Solution: **adjust ϵ** to compensate for these differences

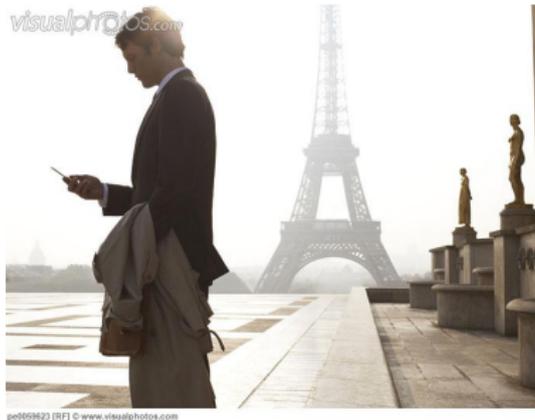
$$\epsilon' = \epsilon + \frac{1}{u} \ln \frac{q - 2 + 3e^{\epsilon v \sqrt{2}}}{q - 5}$$



Case study: Location-Based Services

Retrieve location-dependent information

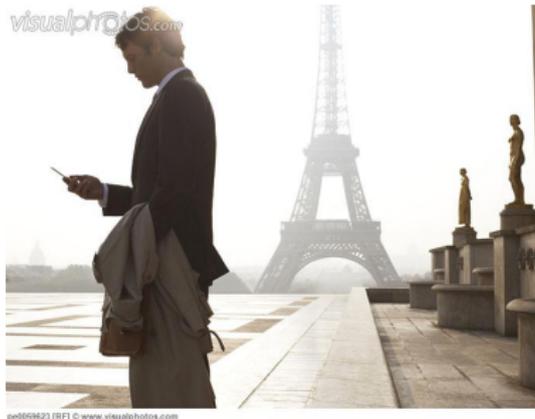
- ▶ Restaurants
- ▶ Friends
- ▶ Gas stations
- ▶ Weather
- ▶ ...



Case study: Location-Based Services

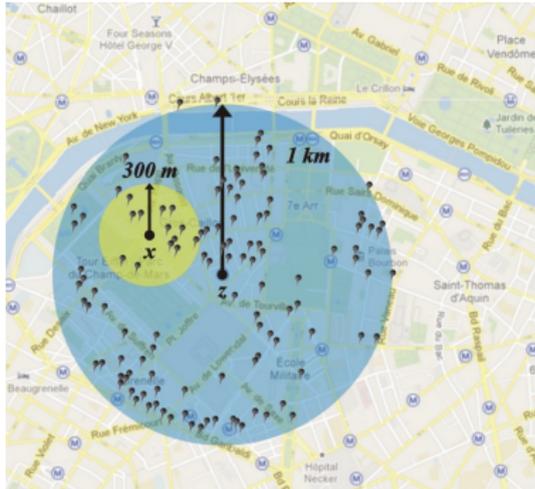
Solution:

- ▶ Add noise to the location x to obtain z
- ▶ Use z to query the provider
- ▶ Some services are insensitive to “small” perturbations (eg. weather, gas stations)
- ▶ In this case the quality of the results will not be affected

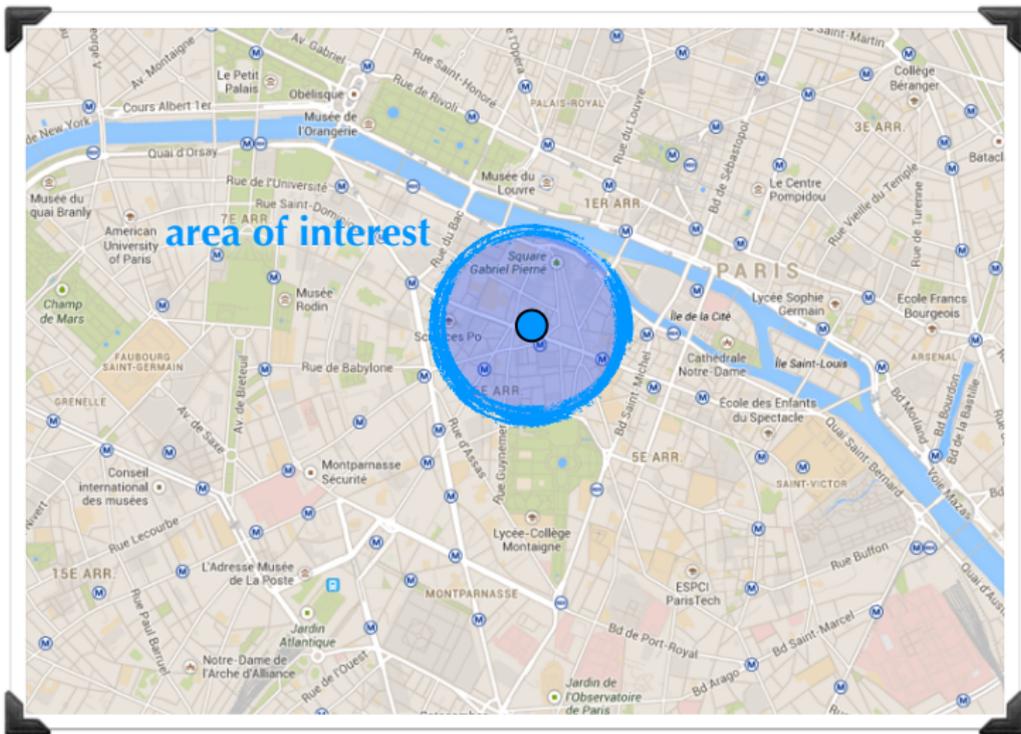


Case study: Location-Based Services

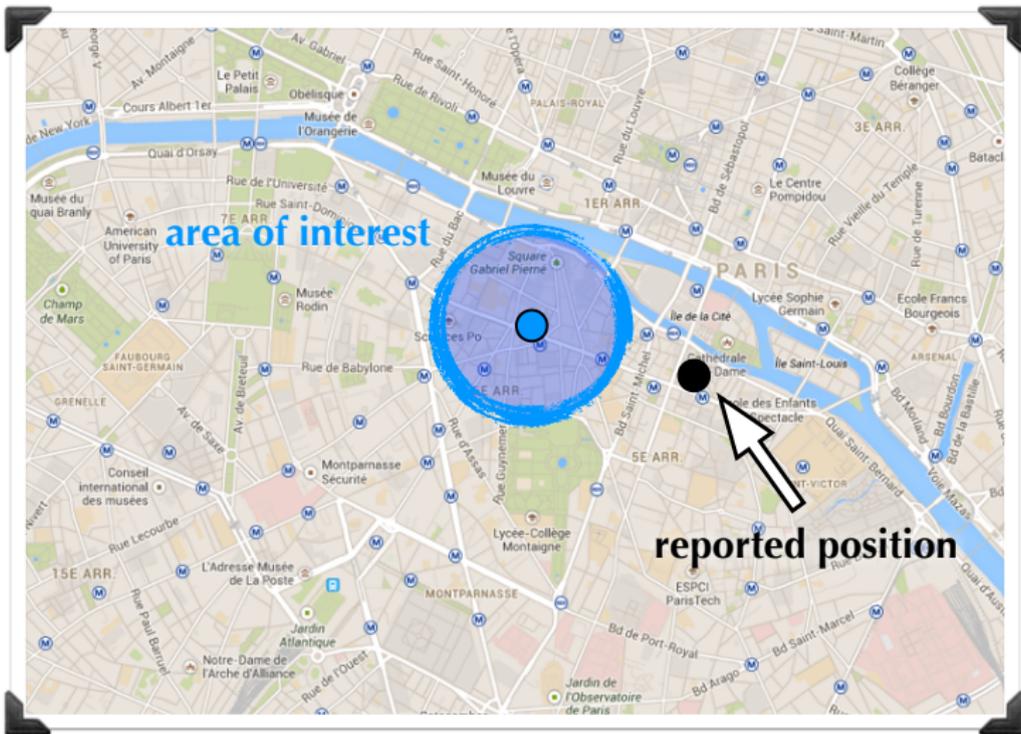
- ▶ Many LBS depend on the accuracy of the location
eg. find restaurants within **300m from x**
- ▶ In this case the query needs to be extended to a larger area
eg. get restaurants within **1km from z**
- ▶ Important: the area needs to be **independent from z**



Obfuscation



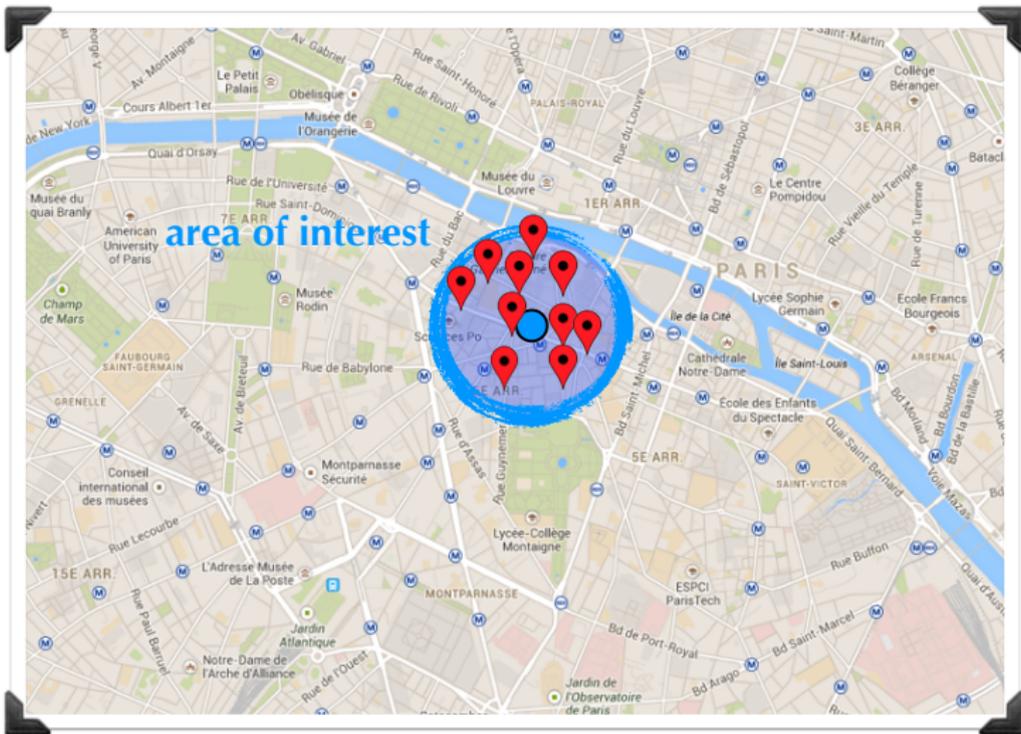
Obfuscation



Obfuscation



Obfuscation



Privacy versus utility: evaluation

- $9 \times 9 = 81$ “points”.
- We compare 4 mechanisms.
- Configured to the same utility.
- Optimal mechanism by [Shroki et al., S&P 2012] for the corresponding prior. Obtained by linear optimization techniques.
- Three prior independent:
 - Planar Laplacian (discretized).
 - Optimal under uniform prior.
 - Simple cloaking.

1	2	3	4	5	6	7	8	9
10	11	12	13	14	15	16	17	18
19	20	21	22	23	24	25	26	27
28	29	30	31	32	33	34	35	36
37	38	39	40	41	42	43	44	45
46	47	48	49	50	51	52	53	54
55	56	57	58	59	60	61	62	63
64	65	66	67	68	69	70	71	72
73	74	75	76	77	78	79	80	81

Privacy versus utility: evaluation

- We fix the utility and measured the privacy.
- Utility loss measured as the **expected distance** between the true location and the reported one [Shroki et al., S&P 2012]
- Privacy measured as the **expected error of the attacker** (using prior information) [Shroki et al., S&P 2012]
- Priors: uniform over colored regions

1	2	3	4	5	6	7	8	9
10	11	12	13	14	15	16	17	18
19	20	21	22	23	24	25	26	27
28	29	30	31	32	33	34	35	36
37	38	39	40	41	42	43	44	45
46	47	48	49	50	51	52	53	54
55	56	57	58	59	60	61	62	63
64	65	66	67	68	69	70	71	72
73	74	75	76	77	78	79	80	81

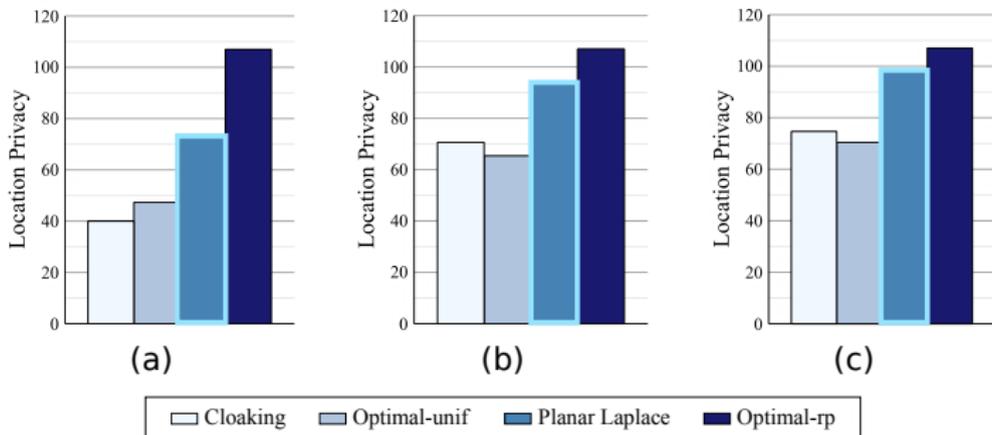
1	2	3	4	5	6	7	8	9
10	11	12	13	14	15	16	17	18
19	20	21	22	23	24	25	26	27
28	29	30	31	32	33	34	35	36
37	38	39	40	41	42	43	44	45
46	47	48	49	50	51	52	53	54
55	56	57	58	59	60	61	62	63
64	65	66	67	68	69	70	71	72
73	74	75	76	77	78	79	80	81

1	2	3	4	5	6	7	8	9
10	11	12	13	14	15	16	17	18
19	20	21	22	23	24	25	26	27
28	29	30	31	32	33	34	35	36
37	38	39	40	41	42	43	44	45
46	47	48	49	50	51	52	53	54
55	56	57	58	59	60	61	62	63
64	65	66	67	68	69	70	71	72
73	74	75	76	77	78	79	80	81

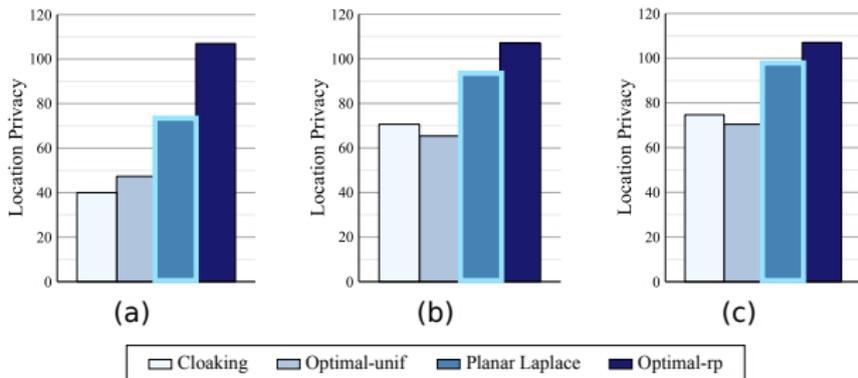
Privacy versus utility: evaluation

The four mechanisms:

- Cloaking,
- Optimal by [Shroki et al. S&P 2012] for the uniform prior
- Ours (Planar Laplacian)
- Optimal by [Shroki et al. S&P 2012] for the given prior



Privacy versus utility: evaluation



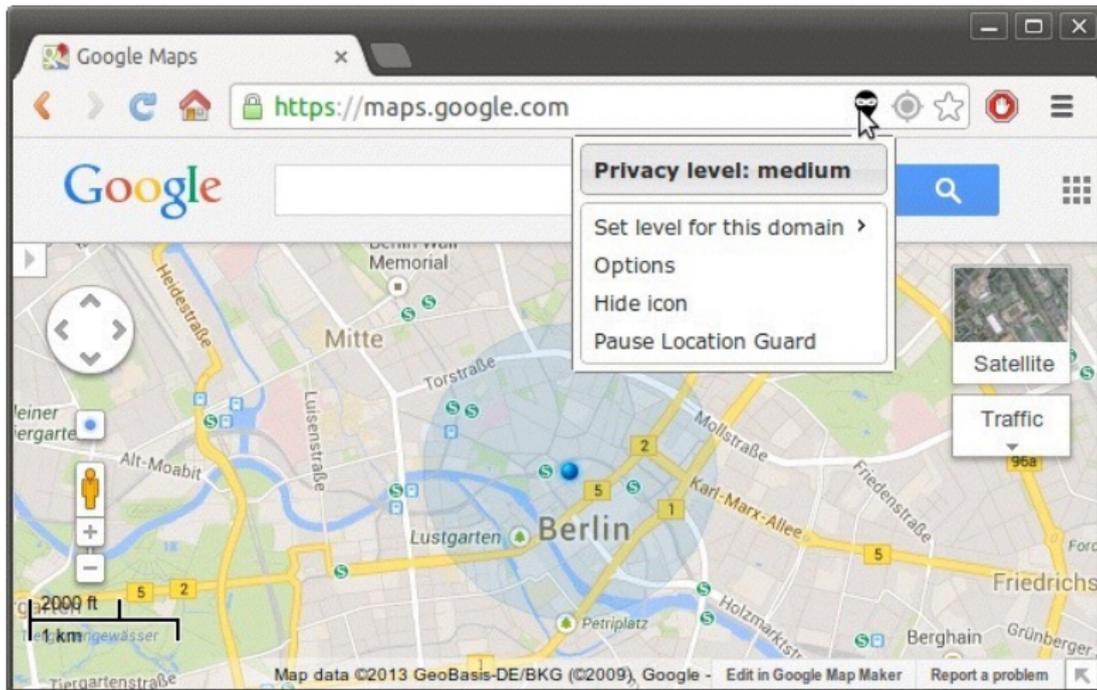
With respect to the privacy measures proposed by [Shokri et al, S&P 2012], our mechanism performs better than the other mechanisms proposed in the literature which are independent from the prior (and therefore from the adversary)

The only mechanism that outperforms ours is the optimal by [Shokri et al, S&P 2012] for the given prior, but that mechanism is adversary-dependent

Tool: "Location Guard"

<http://www.lix.polytechnique.fr/~kostas/software.html>

About 50,000 active users to date



Location Guard: goals

Provide a **simple solution**, for **sporadic, real-time** LBS access

Can we make it simple enough so that people **actually use it**?

Understandable, configurable by human beings

Low-level, **application-agnostic** solution

Location Guard: goals

Provide a **simple solution**, for **sporadic, real-time** LBS access

Can we make it simple enough so that people **actually use it**?

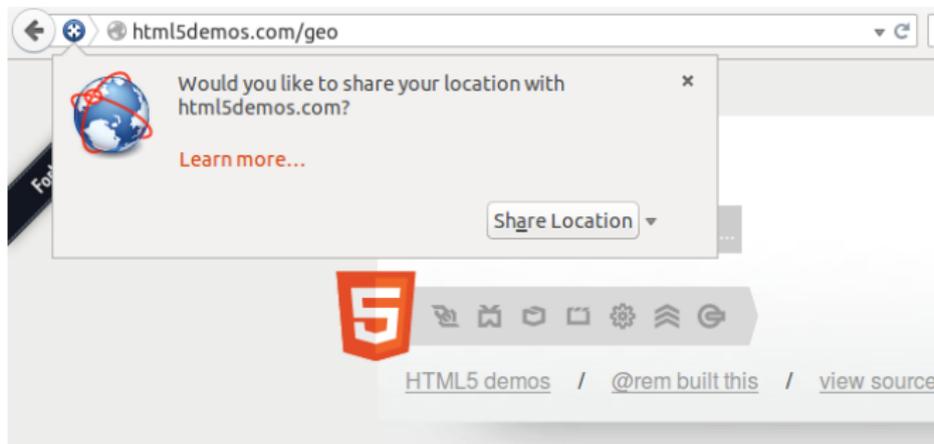
Understandable, configurable by human beings

Low-level, **application-agnostic** solution

OS-level on smartphones (problem: **rooting** the phone)

Browser level (desktop & mobile)

HTML5 geo-location API



Asking the browser for the user's location

Location Guard: adding noise

```
navigator.geolocation.getCurrentPosition(function(pos)
    alert(
        "Latitude: " + pos.coords.latitude +
        "Longitude:" + pos.coords.longitude
    );
);
```

Location Guard: adding noise

```
navigator.geolocation.getCurrentPosition(function(pos)
    alert(
        "Latitude: " + pos.coords.latitude +
        "Longitude:" + pos.coords.longitude
    );
);
```

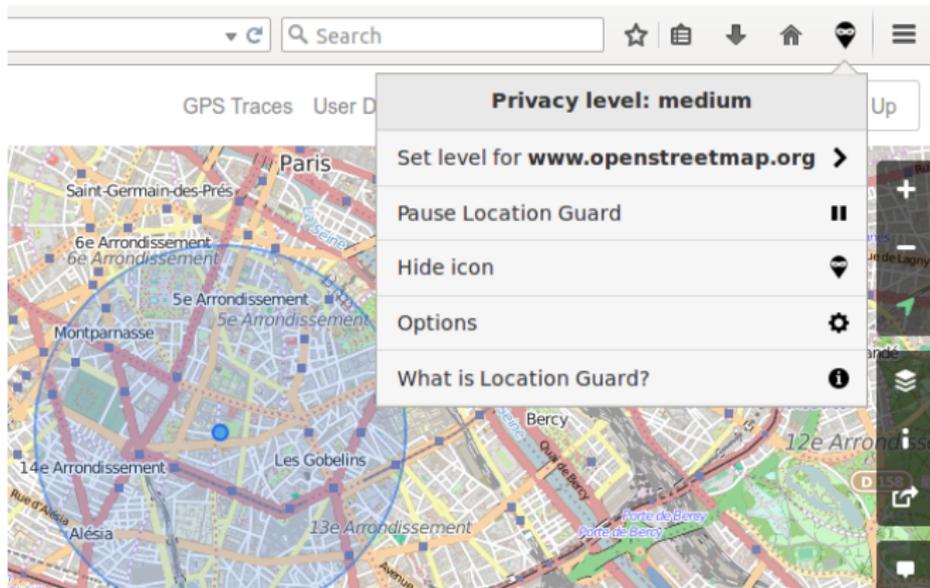
Intercept the javascript call

Content-script, running in separate javascript environment

Inject code in the page, replace navigator.geolocation

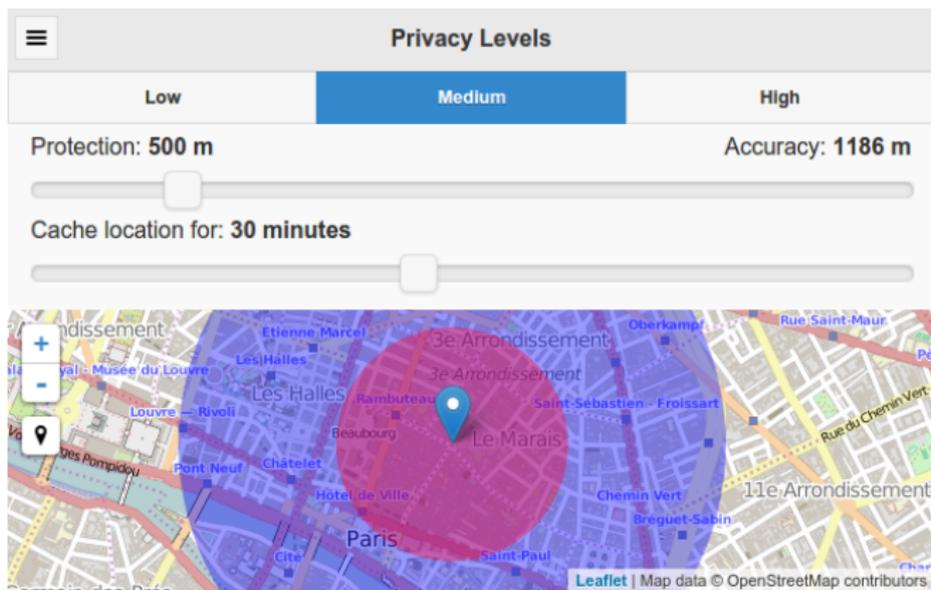
Transparent to the user

User interfaces are hard



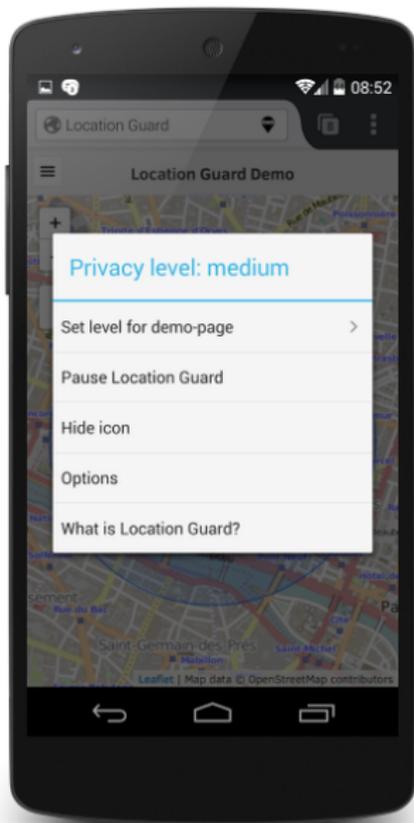
No initial setup, user configuration if needed

User interfaces are hard

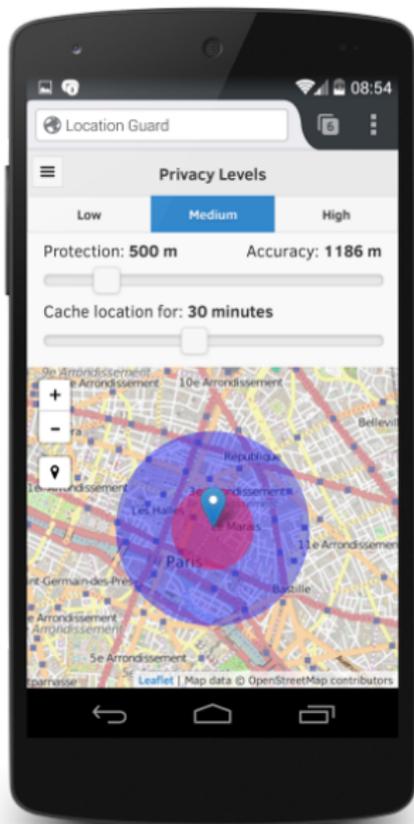


No initial setup, user configuration if needed

Mobile support



Mobile support



User adoption

Timeline

Nov 2013: Chrome

Jul 2014: Firefox

Feb 2015: Firefox Mobile

Feb 2015: Opera

User adoption

Timeline

Nov 2013: Chrome

Jul 2014: Firefox

Feb 2015: Firefox Mobile

Feb 2015: Opera

Current users

Chrome: 6224 active

Firefox: 4642 active

Firefox Mobile: 370 active

Opera: 2134 downloads

How do users discover Location Guard?

No publicity

Huge number of extensions (1367 in Firefox **privacy** category alone!)

How do users discover Location Guard?

No publicity

Huge number of extensions (1367 in Firefox **privacy** category alone!)

Occasional promotion by Google/Mozilla

Mostly by **searching** (users care about privacy!)

“location”: position 1-2

“privacy”: position 35-40

ghacks.net article

You are here: [Home](#) > [Firefox](#) > [Change your location in Firefox using Location Guard](#)

Change your location in Firefox using Location Guard

by [Martin Brinkmann](#) on December 1, 2014 in [Firefox](#) - Last Update: December 1, 2014

Geolocation, the retrieval of a connecting user's location in the world, can be beneficial to both user and website operator. When you connect to a weather website for instance, it is often the case that you want weather information for the location you are at.

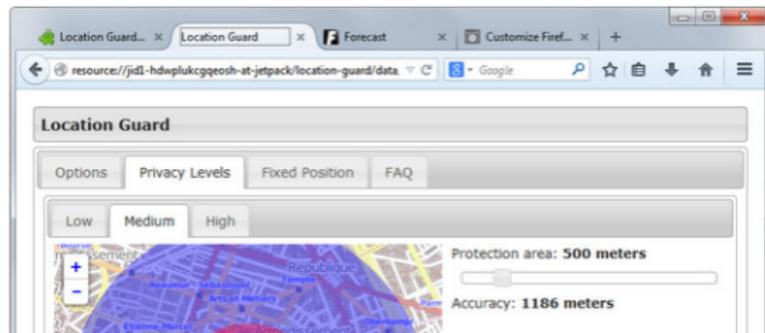
It can also be comfortable to be redirected to a country-specific version of a website.

Sometimes however, geolocation can backfire. This can be the case if you want to look up weather information for another location, if you don't want to be redirected automatically, or if the location that the service discovers is incorrect.

Privacy is usually not part of the problem and the main reason for that is that browsers such as Firefox display prompts before websites may access your location. Then again, if you allow it you may dislike that it can pinpoint your location precisely.

Location Guard is a relative new extension for the Firefox web browser that can be configured individually for each domain you visit that wants to access location-based features.

It offers two main features: the first enables you to add noise to your location so that it cannot be pinpointed with accuracy anymore. The second feature on the other hand sets your location to any place in the world.



Support Us



This site is funded by readers like you. [Support our community on Patreon](#). Even \$1 helps!

Want to make a one time donation instead? [Find out more about it here](#).

POPULAR

LATEST

Ghacks is dying and needs your help

FEBRUARY 27, 2015

Will you make the move to Windows 10?

FEBRUARY 3, 2015

Fix Add-ons not working in Firefox 35

JANUARY 18, 2015

Subscribe / Connect

Subscribe to our newsletter, RSS, or follow us on Facebook, Twitter or Google+.



ghacks Technology Newsletter

Your Email Address

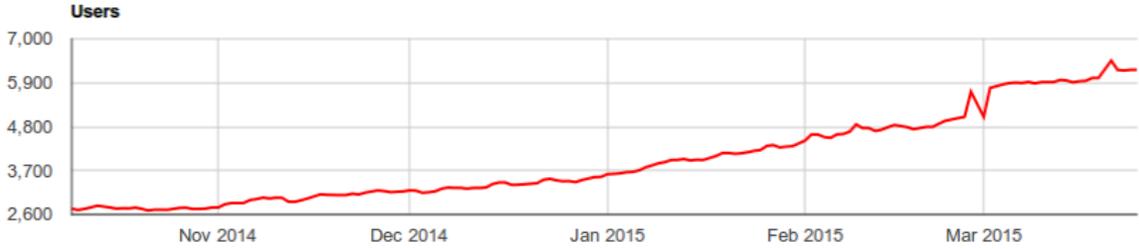
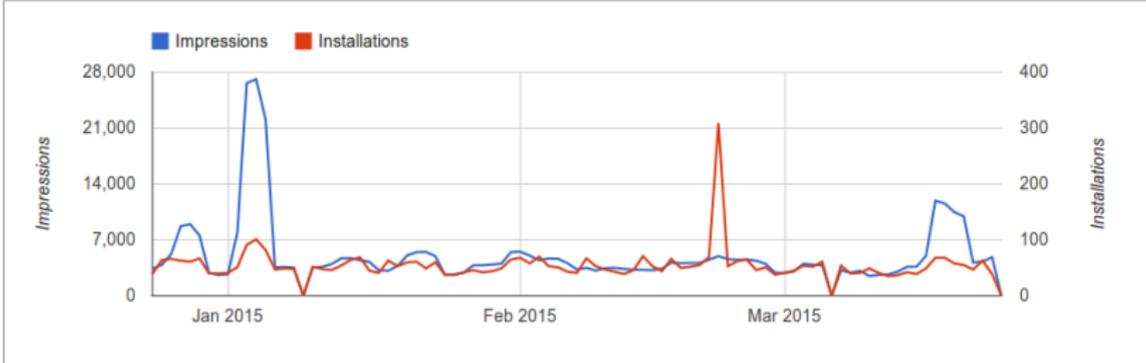
Advertisement

[We Need Your Help](#)

ghacks.net article



Chrome: linear growth



Questions?