Location-privacy in the mobile era: challenges and solutions

Bologna, 18 December 2017

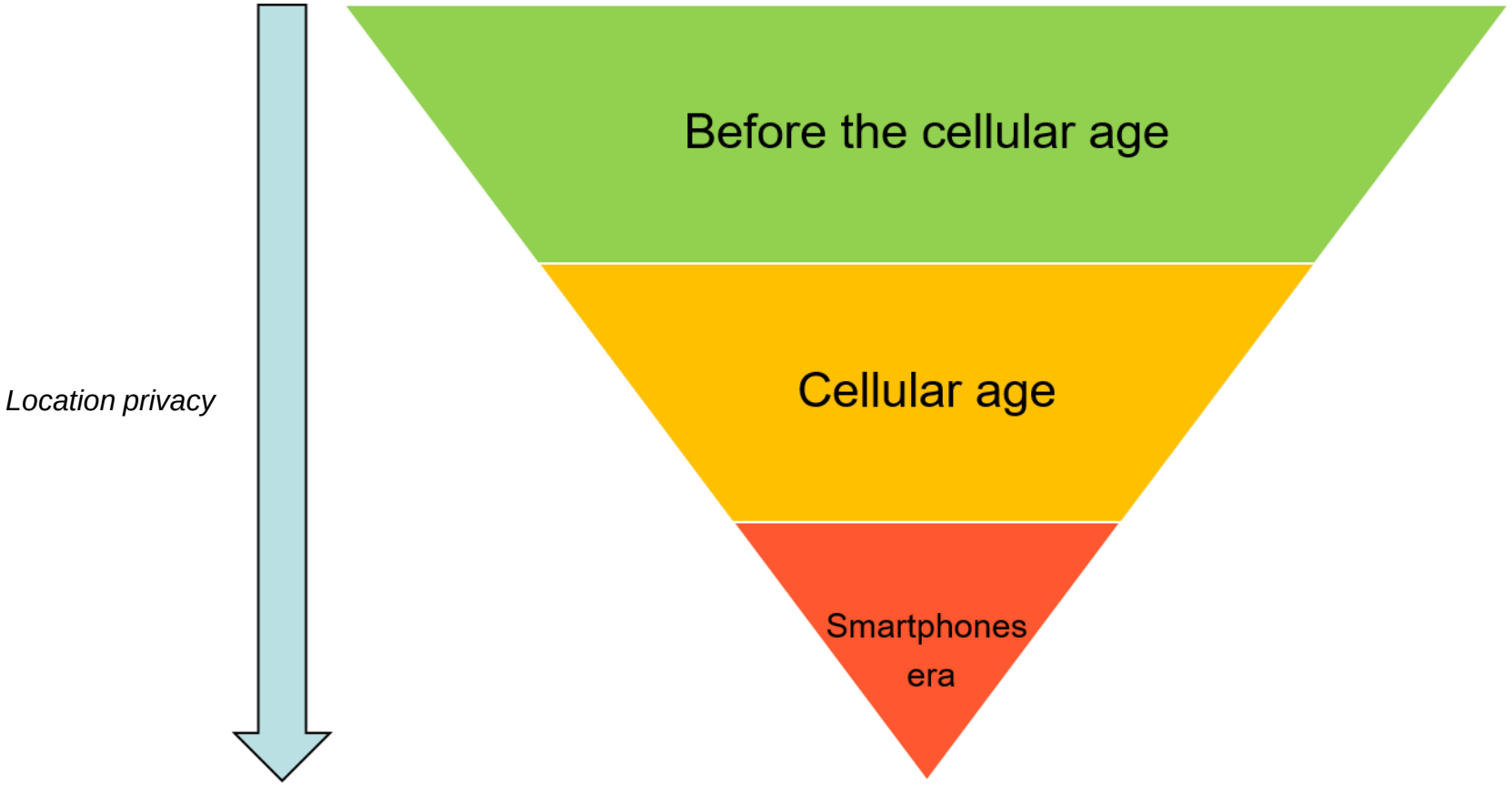
Luca Calderoni

Location privacy

"Location privacy refers to the the ability of an individual to move in public space with the reasonable expectation that their location will not be systematically and secretly recorded for later use."

Andrew J. Blumberg - Stanford University

Location privacy timeline



Location-based services

Explicit

Urban surveillance, monitoring systems, military applications

This services, commonly used for defense and internal security, require the user to disclose its position explicitly. Anyway, they could be designed in order not to use positional data when it is not strictly needed. Moreover, the provider side need some privacy guarantees as well.

Implicit

These services provide the user with some kind of information and collect his location as a hidden side effect. Users focus their attention on the final result and are not aware of the data flow disclosed to the provider.

Location sensing (1)

What events may threat our location privacy around us …

- Smart cards and swipe-cards for integrated mobility services
- Electronic tolling devices (FastTrak, EZpass, congestion pricing)
- E-passports and EMRTD in general
- Services telling you when your friends are nearby
- Free Wi-Fi with ads for businesses near the network access point you're using
- Electronic badges for access control
- Face recognition systems for urban surveillance
- Sensors networks deployed all over the urban and suburban context able to provide the user with some service when combined with his car or with some form of wearable device.

Location sensing (2)

… and on our cell phone

Satellite based

GPS, GLONASS, COMPASS, GALILEO, QUASI-ZENITH

Radio-frequency based

Wi-Fi, Bluetooth, RFID, NFC

Sensor based

Accelerometer, Magnetometer, Gyroscope

Mobile provider based

Cell Towers, Cells ID

Location-based threats

La lack of location privacy may conceal several drawbacks.

Who takes control of our positional database is able to answer the following questions:

- Were you near the house of one “John Doe” last night?
- Did you walk into an abortion clinic?
- Have you been checking into a motel at lunchtimes?
- Why was your secretary with you?
- Which church or mosque do you attend?
- Which gay bars?
- Which political party site are you used to visit?
- Who is my ex-girlfriend going to dinner with?
- …

Combine anonymized positions with named records (1)

"The continued accumulation of location data may reach a point where a marketer can uniquely match an anonymous location trace to a named record in a separate database"

Stephen B. Wicker – Cornell University

Combine anonymized positions with named records (2)

Named records containing user's preferences

$$\mathbf{x}_i = (x_{i,0}, x_{i,1}, x_{i,2}, \ldots, x_{i,n-1});\ x_{i,j} \in \{0, 1, e\}$$

Anonymous location trace

$$\mathbf{L}_m = (l_0, l_1, l_2, \ldots l_{m-1})$$

A mapping function derives preferences from location trace

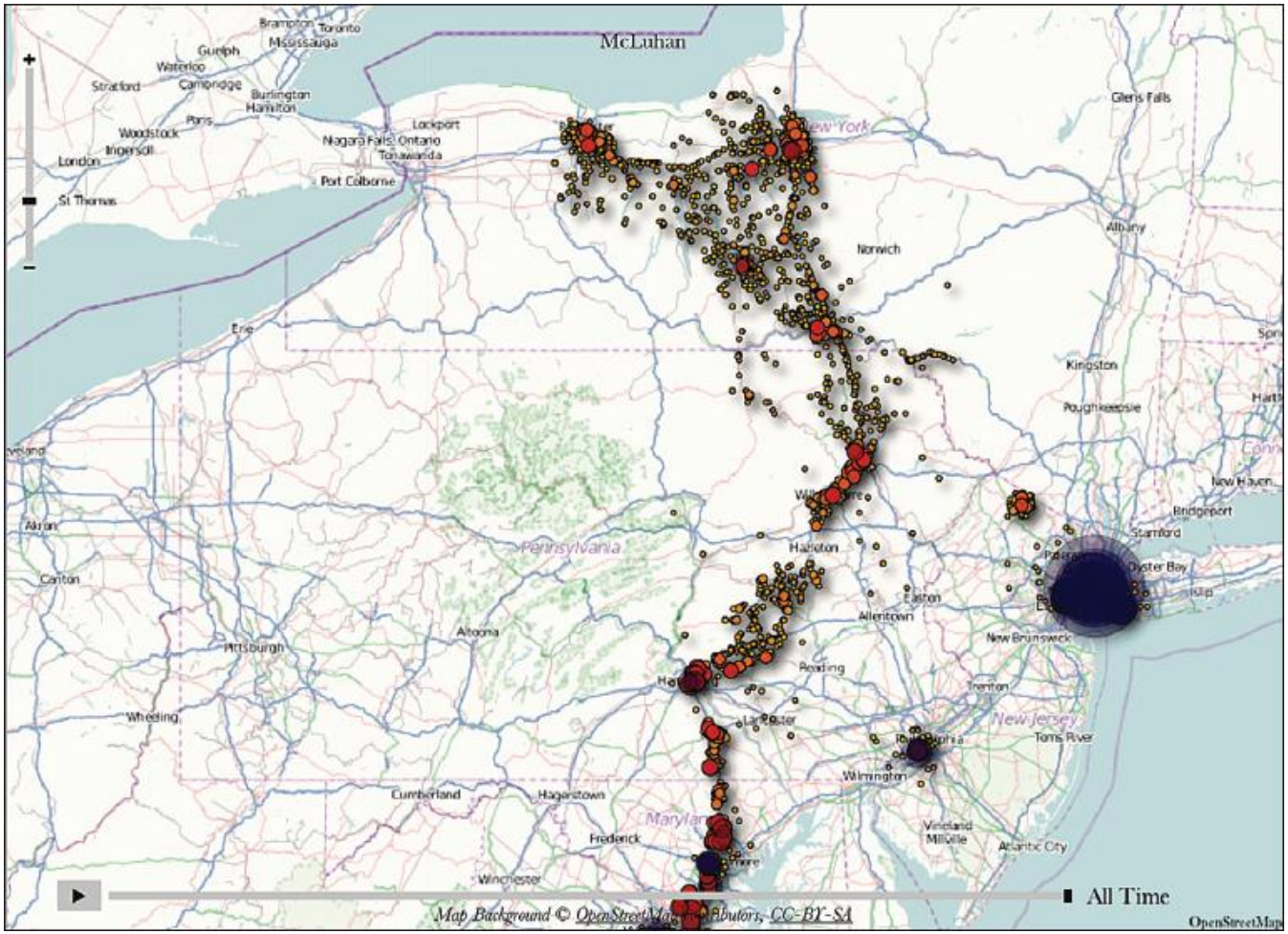
$$F : \{\mathbf{L}_m\} \longrightarrow \{\mathbf{P}\}$$

$$\mathbf{P} = (p_0, p_1, p_2, \ldots, p_{n-1});\ p_j \in \{0, 1, e\}$$

As the length *m* of a location trace $\mathbf{L}_m$ increases, the number of non-erased coordinates of a preference vector **P** increases; the overall effect is an increase in minimum distance and a corresponding increase in the efficacy of correlation attacks.

Case studies: consolidated.db

During 2011, U.K. researchers Alasdair Allan and Peter Warden caused a media frenzy by announcing their discovery of an iPhone file named '*consolidated.db*' that contained a time-stamped location trace of the user.



The consolidated.db location trace. Image credits Stephen B. Wicker.

Case studies: Admiral

During 2016, Admiral, a UK insurance company, planned to analyze the Facebook accounts of first-time car drivers or owners to look for personality traits linked to safe driving. This data, including location traces, may be used to infer the risk and trust related to each customer and adjust the total amount of its insurance fee.

The most insightful point is that customers could be completely unaware of what's going on at the company's employee workstation. They could be simply asked for an higher fee.

Anonymization techniques

Location perturbation

The principal issue here is that user's position is altered, thus, the provided location-based service could be compromised.

Access control

User is supposed to grant access to its positional data to a restricted set of known providers. The problem is we need trusted parties.

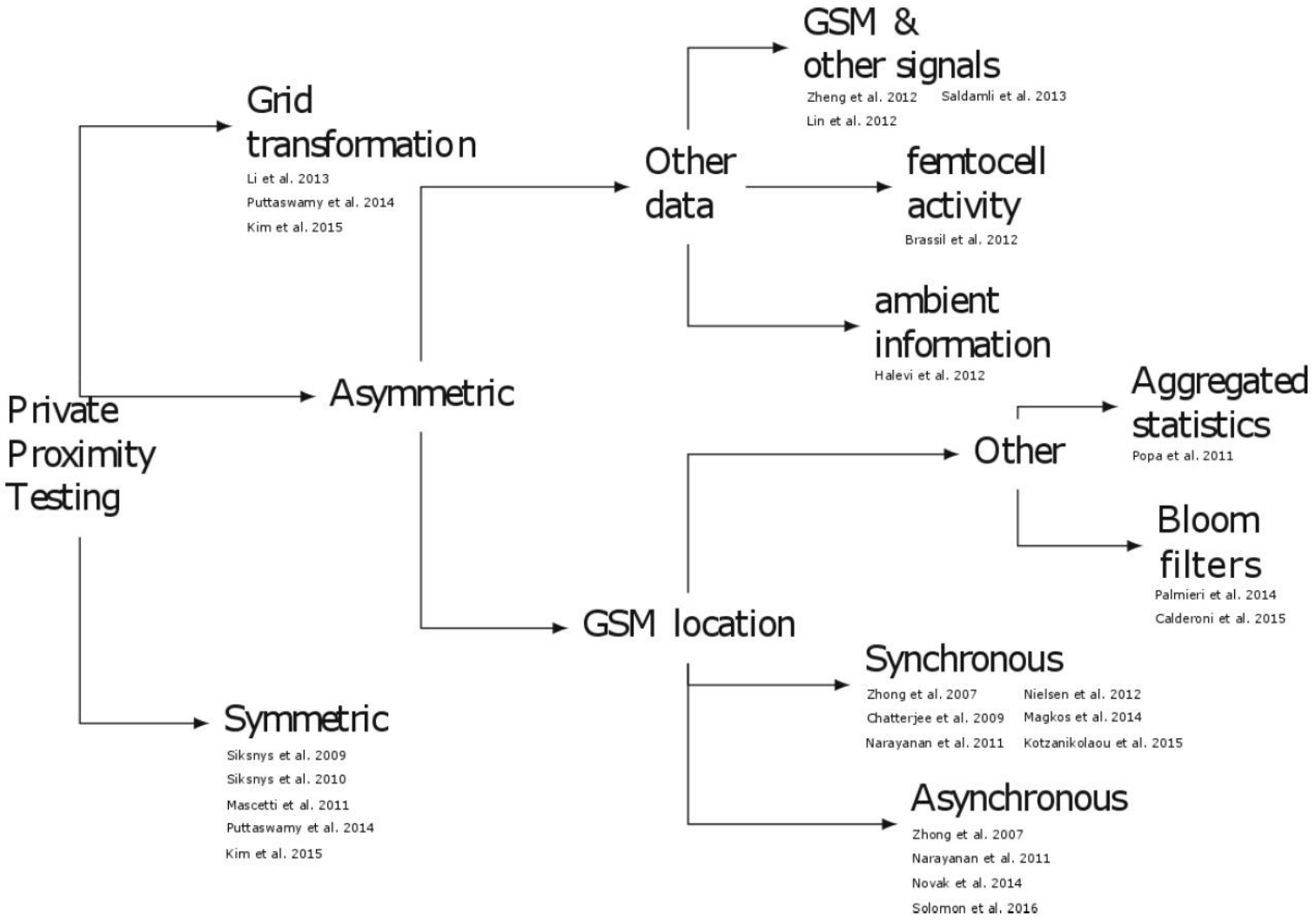
Private information retrieval

The user does not disclose his position at all. He learns some location-based information from the provider side and send specific request on that basis. Some LBS are not suitable for such a setting; moreover, there is no privacy at all for the provider side.

Encryption

A privacy-oriented protocol combined with encryption preserve privacy whithout compromising the service.

Private proximity testing: state of the art



SKNN (1)

This method relies on a protocol which performs several operations in a secure and privacy preserving way in order to compute a *Secure Squared Euclidean Distance*. It was extended by Solomon et al. to deal with location data, i.e. to securely compute the distance between two points.

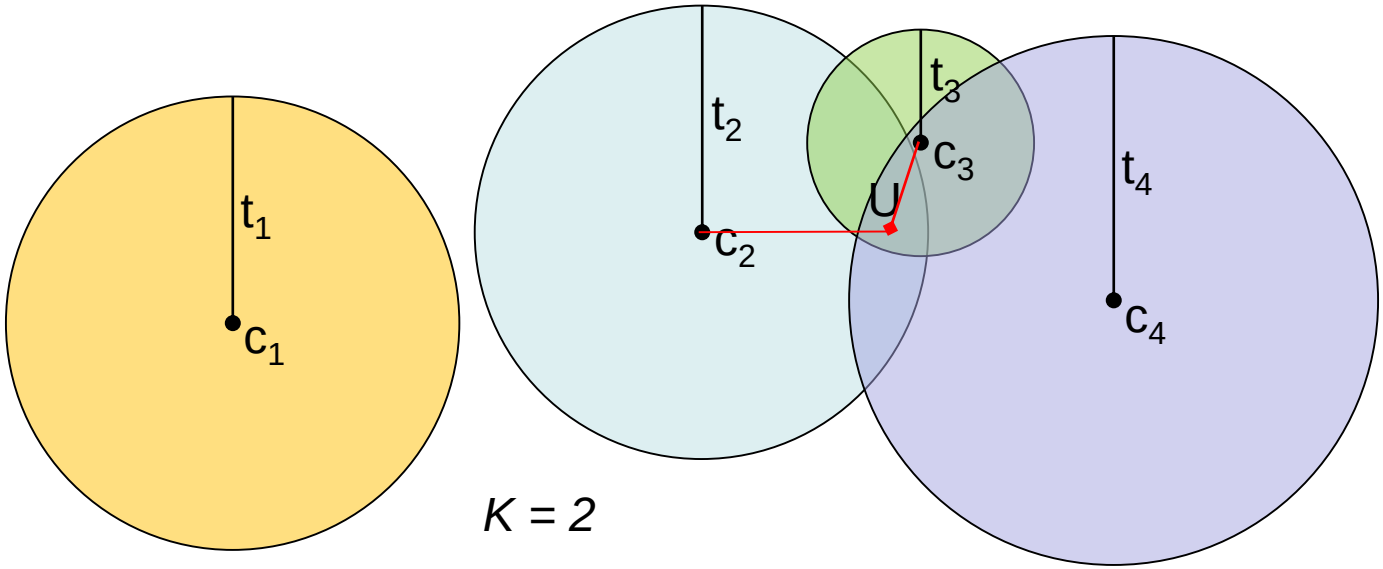
Secure Multiplication + Secure Squared Euclidean Distance + Secure Bit Decomposition + Secure Minimum out of *n* Numbers + Secure Bit-OR

=

Secure K-NearestNeighbor

K-NN + Paillier Cryptosystem

SKNN (2)

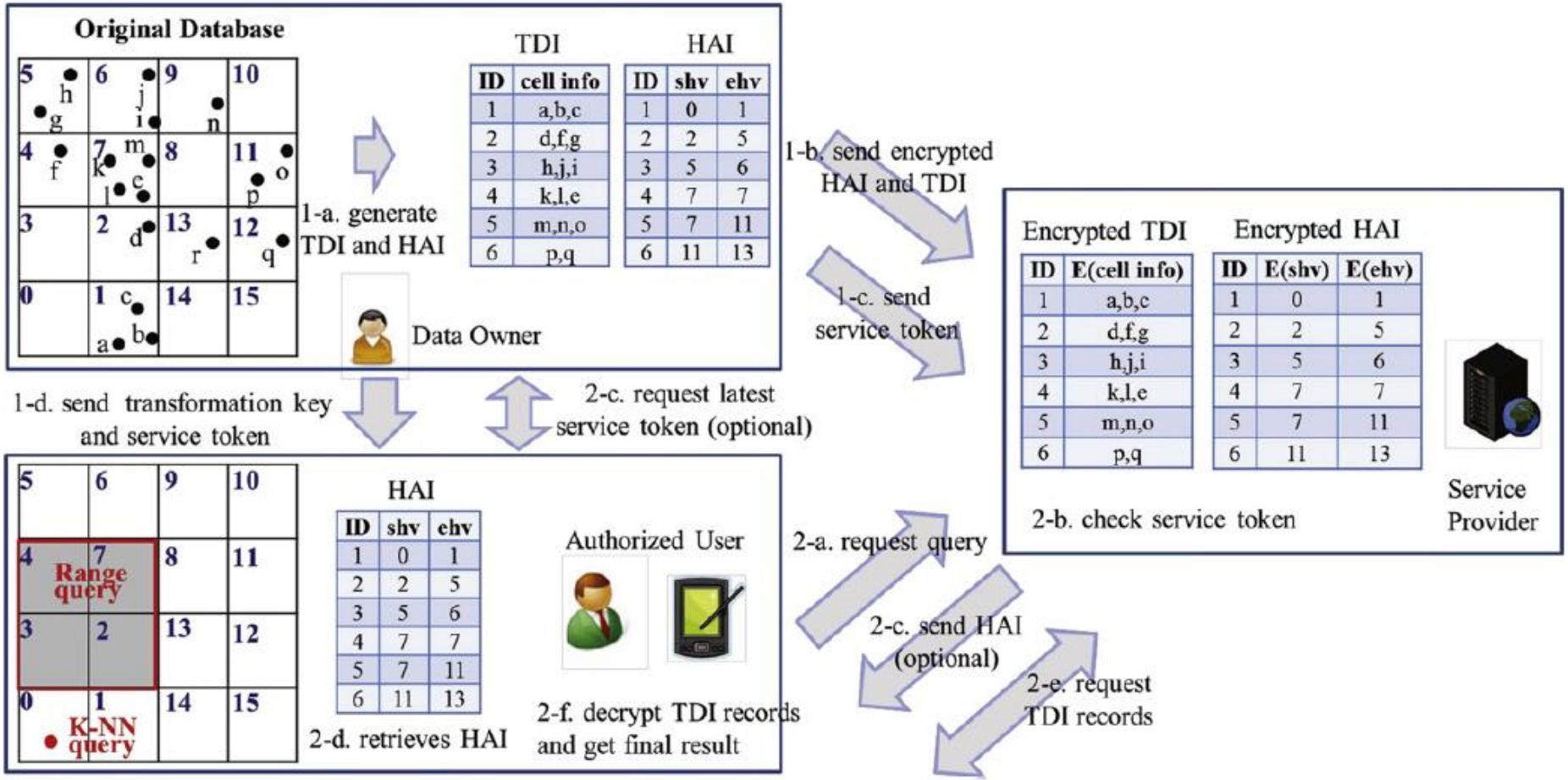


The first *k-nearest* AoI centers are securely returned by the algorithm. As an additional constraint, in order for the AoI_i to be included in the result, the user needs to stay within the distance t_i from the AoI's center.

Problems

- We are allowed to define circular AoI only
- The number of AoI afflicts location-query execution time

Spatial query with Hilbert Curve (1)



Hilbert Curve + AES

Spatial query with Hilbert Curve (2)

- **DO**: defines a conventional grid, computes a Hilbert curve-based transform with the desired Fan-out F and save location data into two tables (TDI, HAI).
- **DO**: encrypts TDI and HAI (with AES) and sends them to the SP. The AES key is shared with all users.
- **SP**: provides users with encrypted TDI and specific records of HAI
- **U**: decrypts TDI and specific records of HAI, applies the Hilbert curve-based transform to perform range queries or k-nn queries.

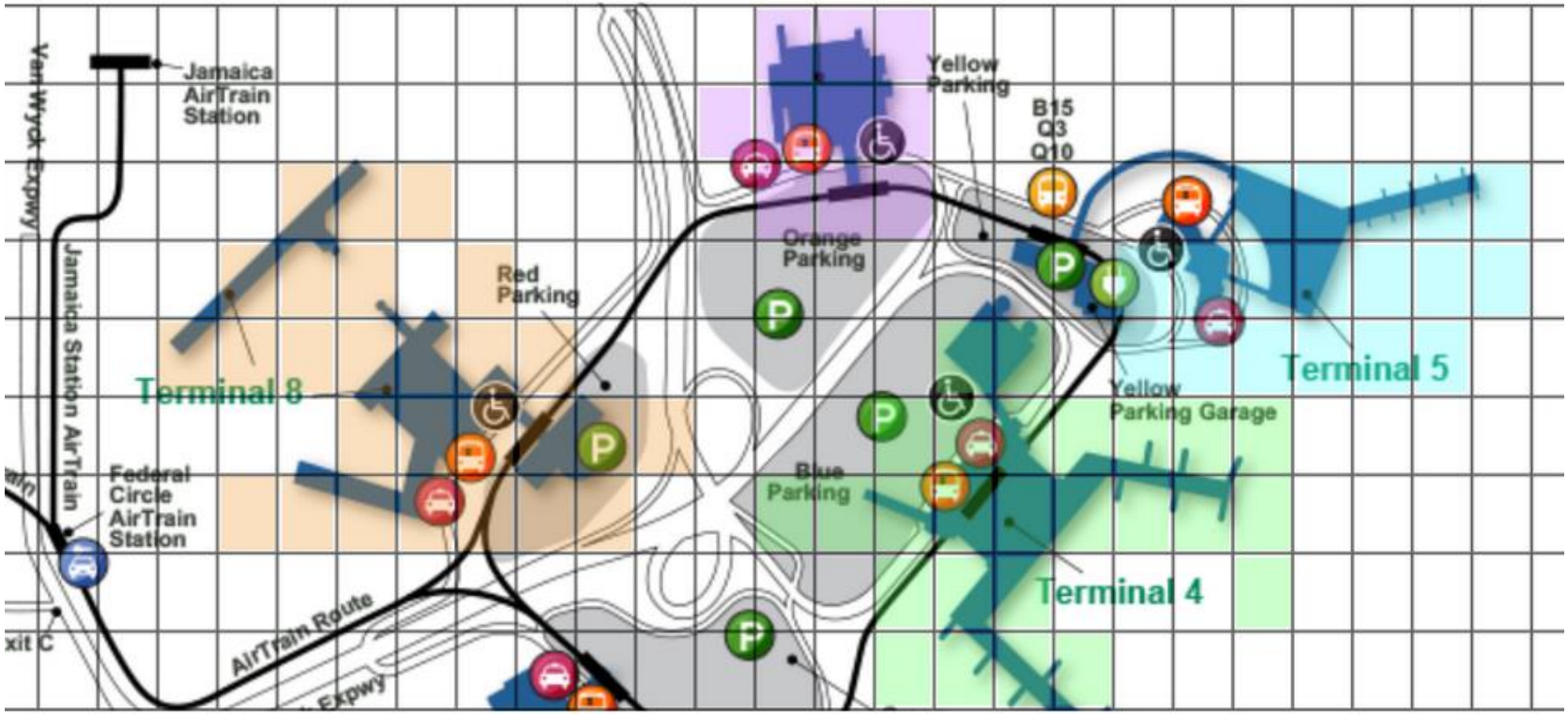
Here the user's position is never sent to the data owner, as the proposed protocol is designed for a three party setting where an untrusted service provider hosts encrypted data sent by the data owner. Hence, this solution is not suitable for certain contexts, such as monitoring service for urban security.

Problems

- Weak security settings (AES, key distribution)
- The user never send his location information at all

SBF

Spatial Bloom Filters rely on the well known Bloom Filter and preserve both user's and provider's privacy. They are combined with the Paillier Cryptosystem.



Bloom Filter + Paillier Cryptosystem

Spatial Representation (1)

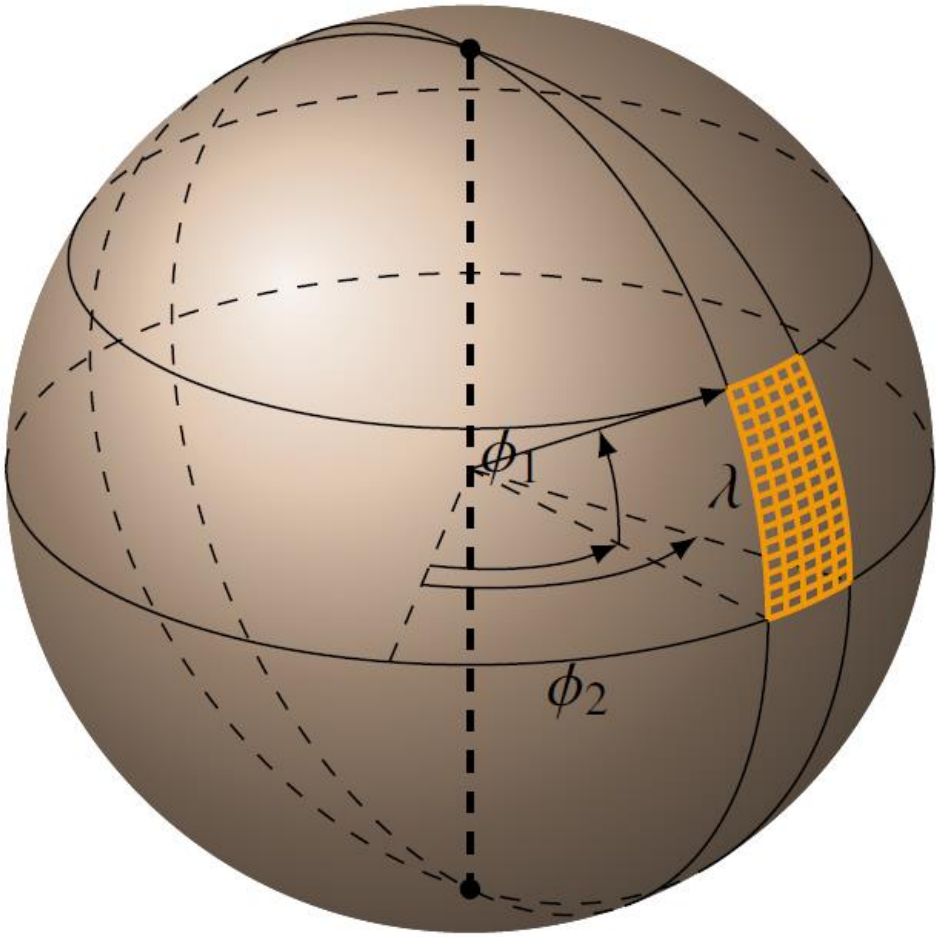
- The precision concerning user's positional data depends on the expected error of the device or sensor used to detect the position.
- In most cases (smartphones and other mobile devices) it's a combination of GPS signal strength and visibility of Wi-Fi networks.
- Current precision is almost address-level, between 10 to 60 meters, depending on many factors.

We define a geographical grid composed of squares of side 0.001 in latitude and 0.001 in longitude

0.001 degrees	lng		lat	
equator	111.32 m	~	111.00 m	
23th parallel N/S	102.47 m	~	111.00 m	Cuba
45th parallel N/S	78.71 m	~	111.00 m	Italy
67th parallel N/S	43.50 m	~	111.00 m	Alaska

Their actual size depends on the position on Earth, but is generally close to 100*m* x 100*m*.

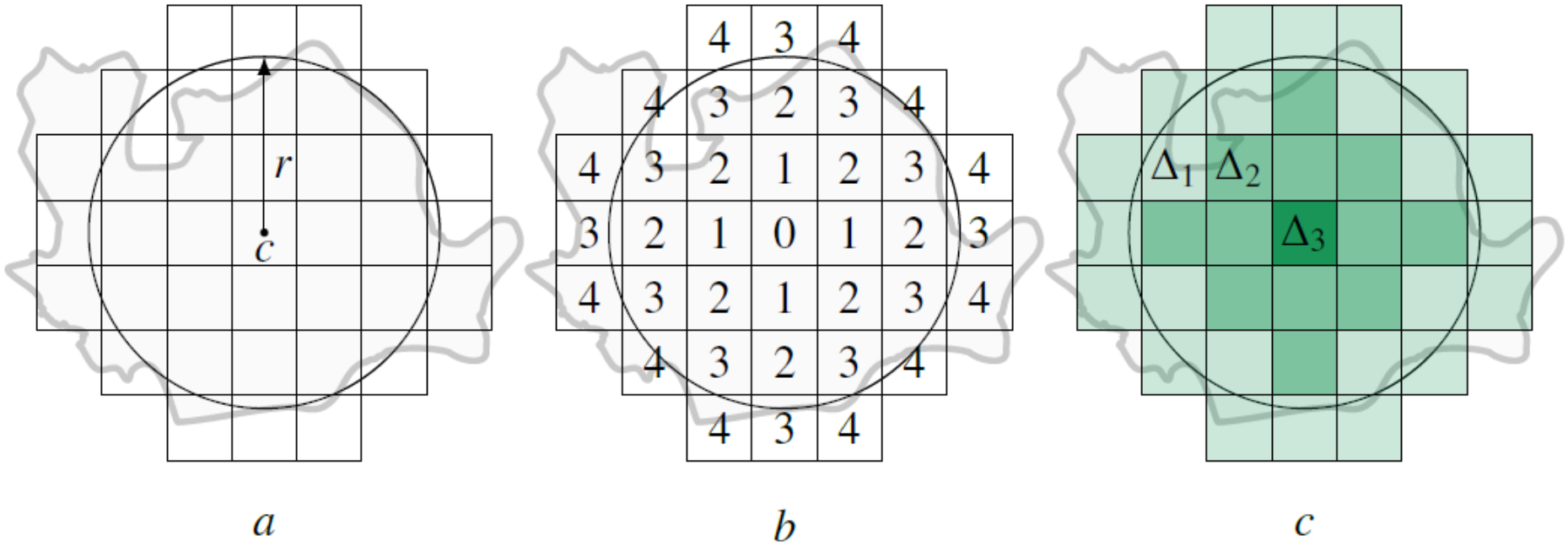
Spatial Representation (2)



An example of the planet's surface and the grid plotted on it. ϕ_1 and ϕ_2 are longitude values while λ is a latitude value.

Spatial Representation (3)

- Let's assume we want to monitor one or more *areas of interest* (AoI)
- As we want to comply with a set-based representation, we consider each AoI as a set of the aforementioned squares.
- NOTE: areas do not need to be concentric nor adjacent.



An example of the area coverage algorithm for concentric Areas of Interest.

Bloom Filters

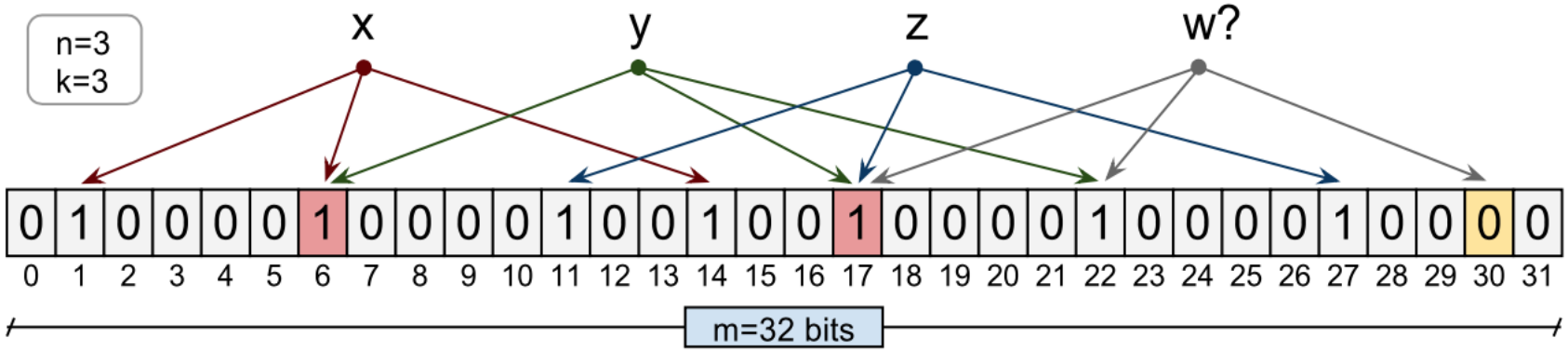
BF is a probabilistic data structure which allows to perform membership queries in an easy and efficient way.

Bloom filter:

- Stores one set
- Affected by false positives
- False negatives are not possible

False positive probability:

$$\left(1 - \left(1 - \frac{1}{m}\right)^{kn}\right)^{k}$$



A classical Bloom filter with 32 bits array. Image credits Tarkoma et al.

Spatial Bloom Filters (1)

s originating sets: $S = \{\Delta_1, \Delta_2, \ldots, \Delta_s\} \quad \bar{S} = \bigcup_{\Delta_i \in S} \Delta_i$

k hash functions: $H = \{h_1, \ldots, h_k\}$

n elements: $|\bar{S}| = n$

m cells: $h_i \in H : \{0,1\}^* \rightarrow \{1, \ldots, m\}$

O (strict total order): $\Delta_i < \Delta_j \; for \; i < j.$

SBF:

$$B^{\#}(S,O) = \bigcup_{i \in I} \langle i, \max L_i \rangle \qquad I = \bigcup_{\delta \in \bar{S}, h \in H} h(\delta)$$

$$L_i = \{l \mid \exists \delta \in \Delta_l, \exists h \in H : h(\delta) = i\}$$

SBF (vector notation):

$$b^{\#}[i] = \begin{cases} l & \text{if } \langle i, l \rangle \in B^{\#}(S,O) \\ 0 & \text{if } \langle i, l \rangle \notin B^{\#}(S,O) \end{cases}$$

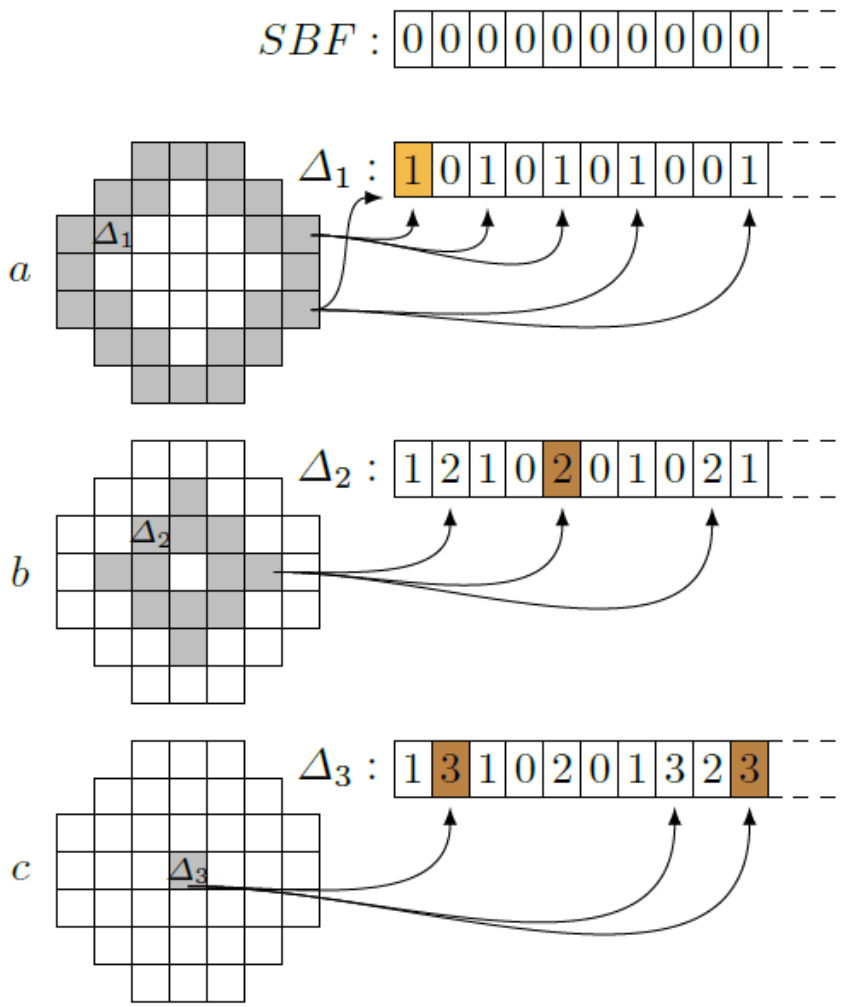
Spatial Bloom Filters (2)

SBF were originally introduced to mutually preserve *location-privacy* for both users and providers using a *location-based* service.

Spatial Bloom filter:

- Stores multiple sets
- Affected by false positives
- False negatives are not possible
- Inter-sets errors

Overwriting rule:
in case of collision, the greater label overwrites the lesser one



Spatial Bloom Filters (3)

SBF construction

Input: $\Delta_1, \Delta_2, \ldots, \Delta_s, H$;
Output: $b^{\#}$;

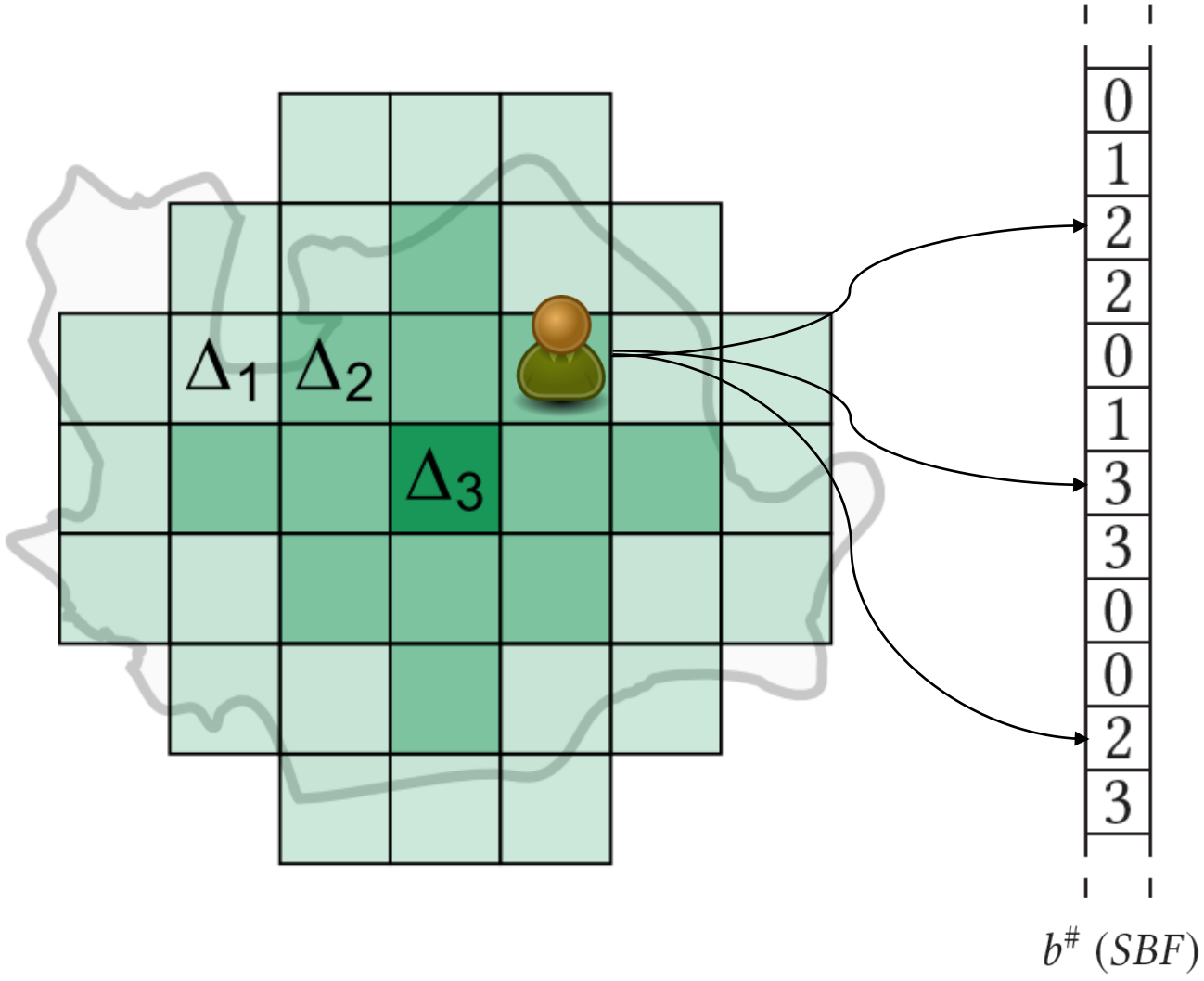
```
1 for i ← 1 to s do
2 |   foreach δ ∈ Δ_i do
3 |   |   foreach h ∈ H do
4 |   |   |   b^# [h(δ)] ← i;
  |   |   end
  |   end
  end
5 return b^#;
```

SBF verification

Input: $b^{\#}, H, \delta_u, s$;
Output: Δ_i;

```
1 i = s;
2 foreach h ∈ H do
3 |   if b^# [h(δ_u)] = 0 then
4 |   |   return false;
  |   else
5 |   |   if b^# [h(δ_u)] < i then
6 |   |   |   i ← b^# [h(δ_u)];
  |   |   end
  |   end
  end
7 return Δ_i;
```

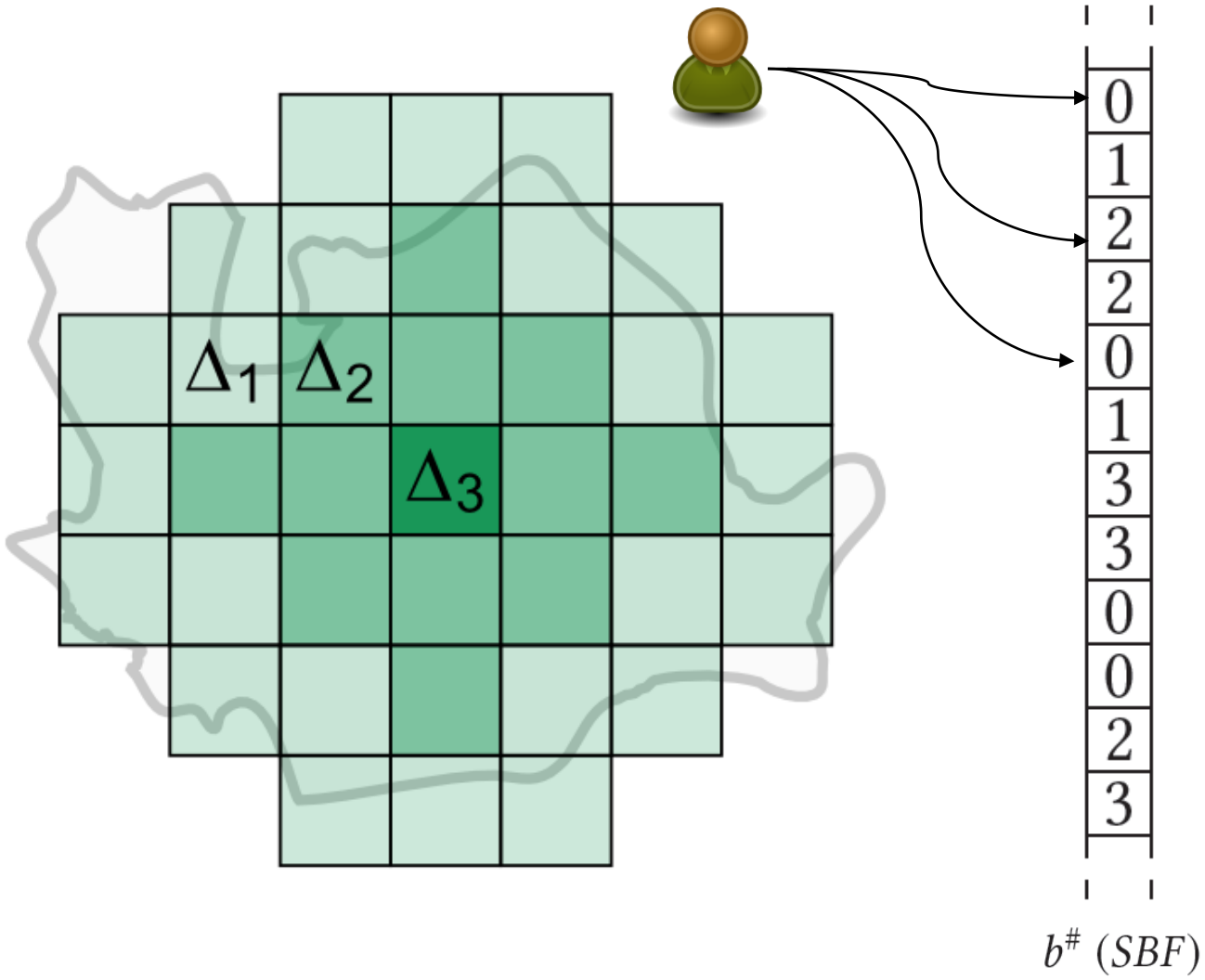
Example: true positive



Response:

User is inside area 2

Example: true negative



Response:

User is outside any area

False positives

SBF are subject to subset-specific false positives probabilities.

Let n_i be the number of members of the set Δ_i :

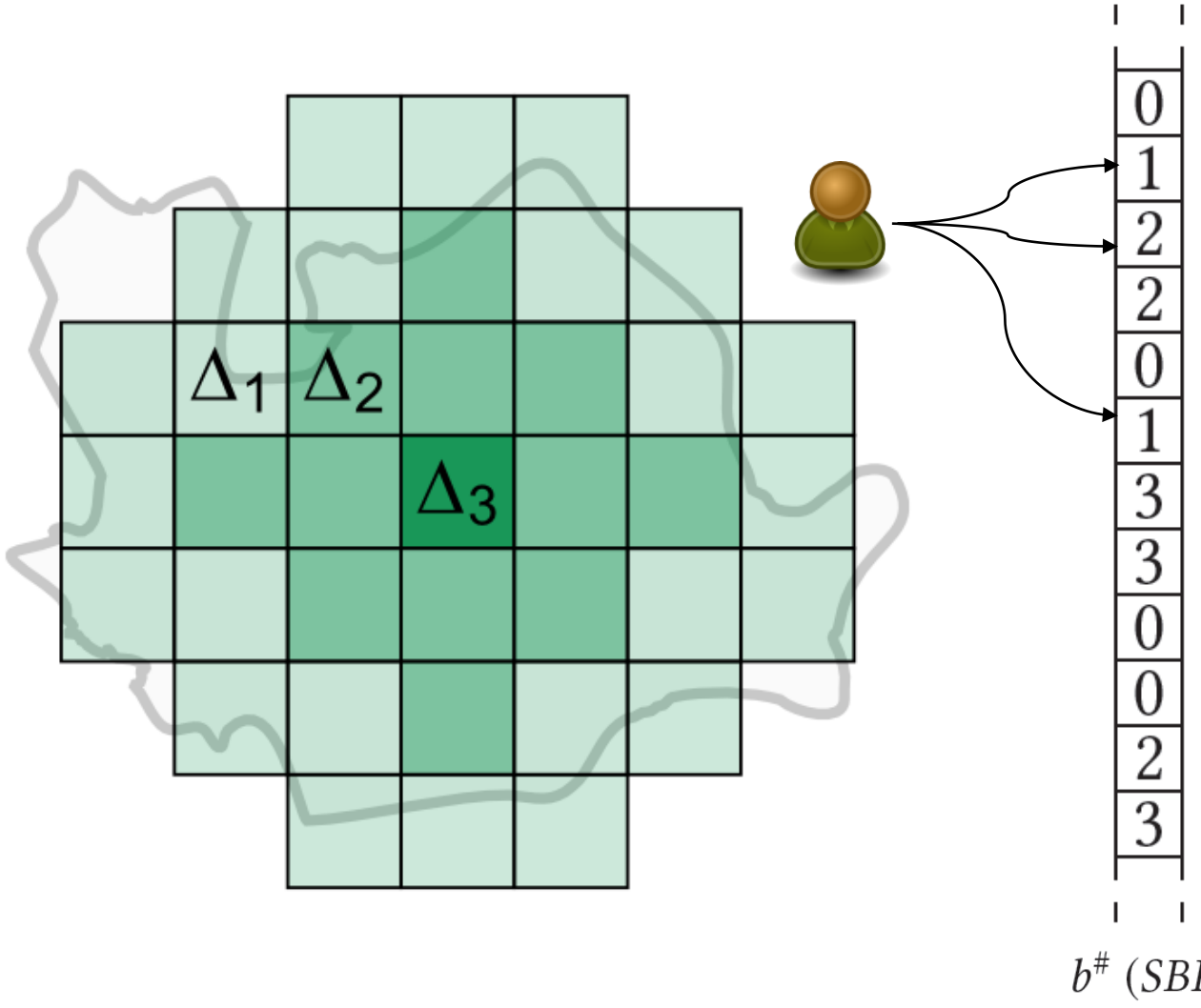
$$fpp_s = \left(1 - \left(1 - \frac{1}{m}\right)^{kn_s}\right)^k$$

$$fpp_{s-1} = \left(1 - \left(1 - \frac{1}{m}\right)^{k(n_s+n_{s-1})}\right)^k - fpp_s$$

$$fpp_1 = \left(1 - \left(1 - \frac{1}{m}\right)^{kn}\right)^k - fpp_s - \cdots - fpp_2$$

$$fpp_i = \left(1 - \left(1 - \frac{1}{m}\right)^{k\sum_{j=i}^{s} n_j}\right)^k - \sum_{j=i+1}^{s} fpp_j$$

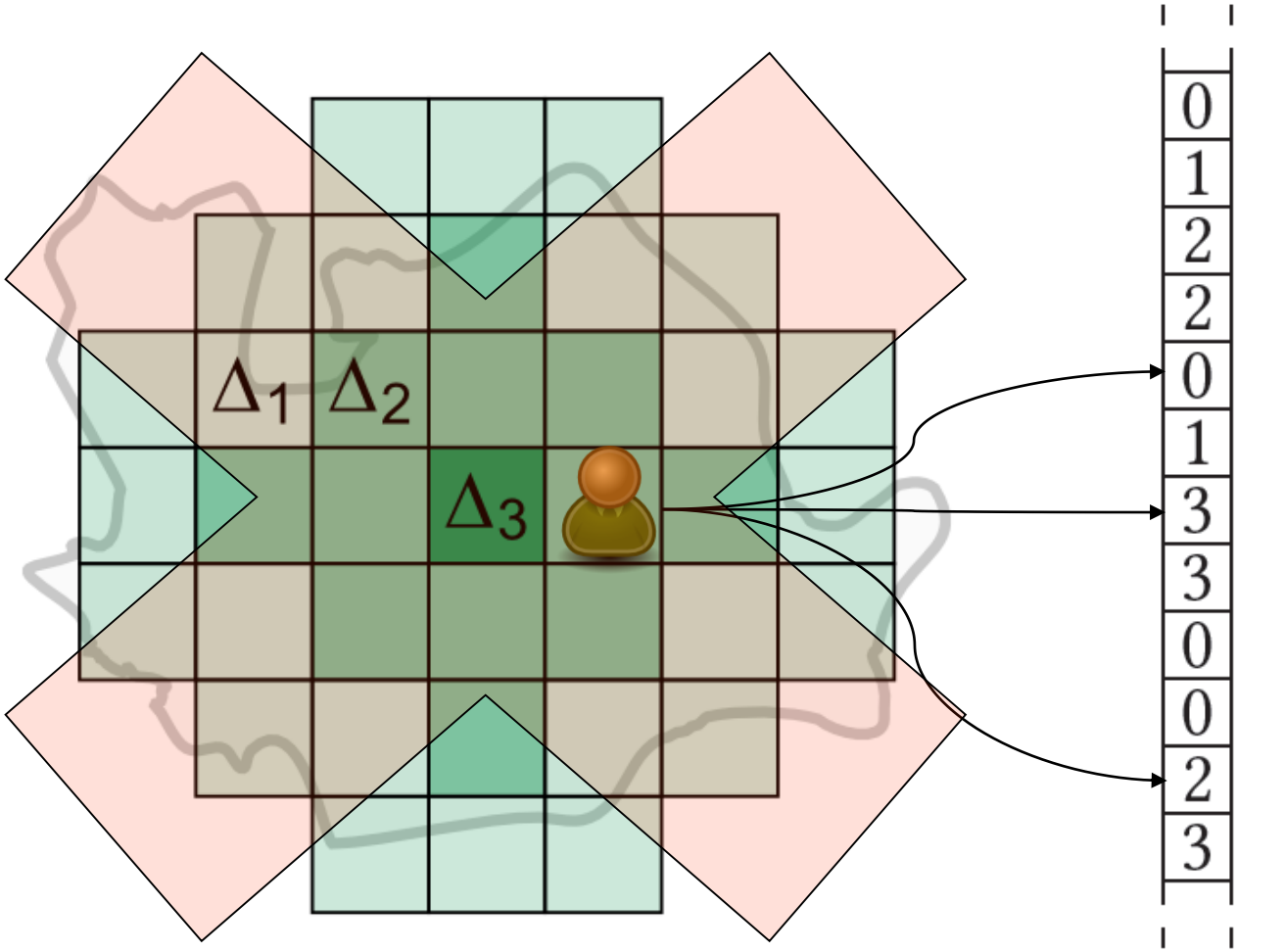
Example: false positive



Response:

User is inside area 1

Example: false negative



$b^{\#}\ (SBF)$

Response:

User is outside any area

<u>May never occur!</u>

Inter-sets errors

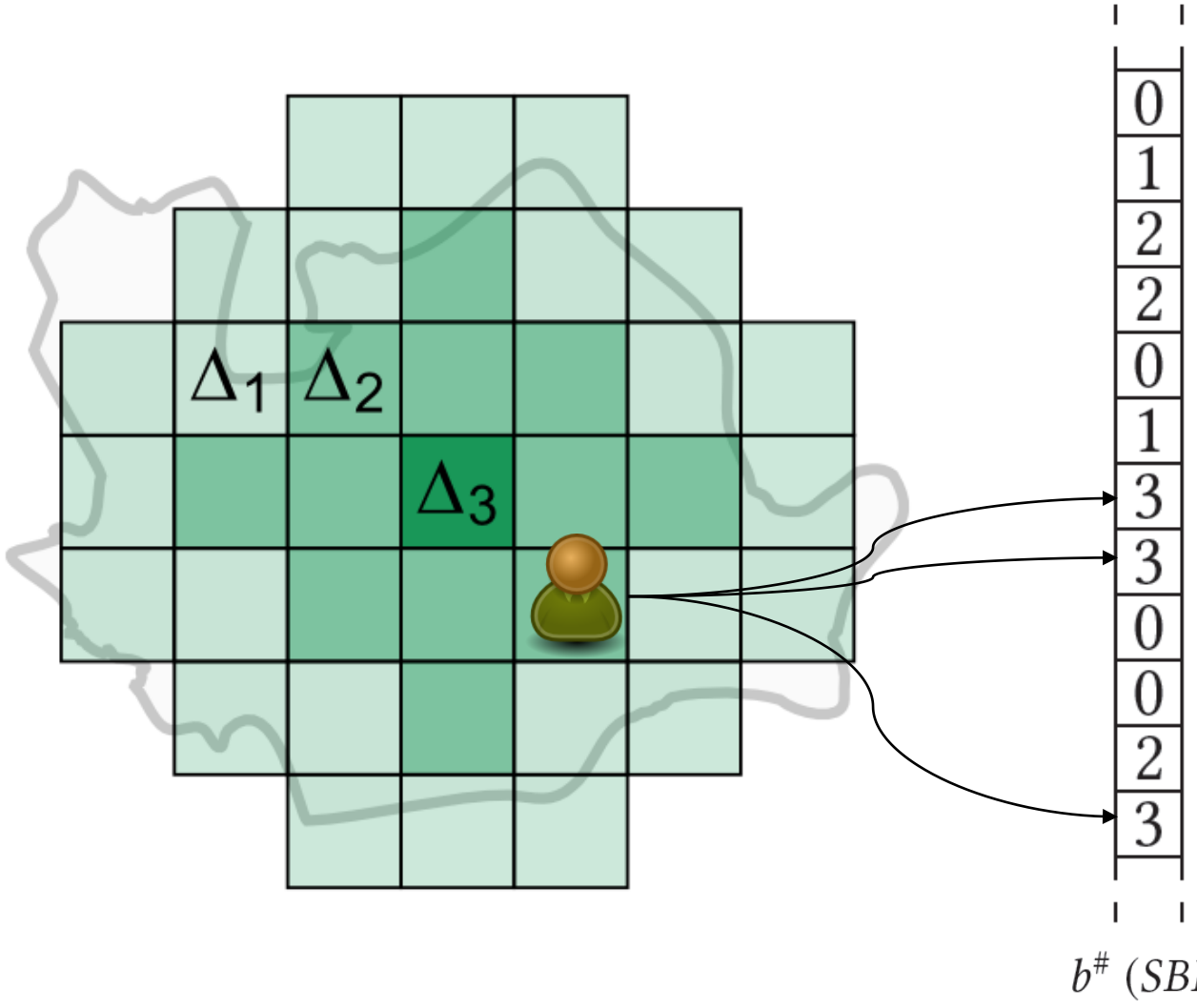
- SBF are affected by *inter-set errors*. This event occurs when an element belonging to a set *i* is wrongly recognised by the SBF as belonging to another set *j* (due to hash collisions and the overwriting rule).

- Fortunately, a SBF can be constructed in order not to produce inter-sets errors at all.

Future research:

We are working on this subject in order to refine the probabilistic model behind the SBF, concerning false positives and inter-set errors. Further results will be published in the near future.

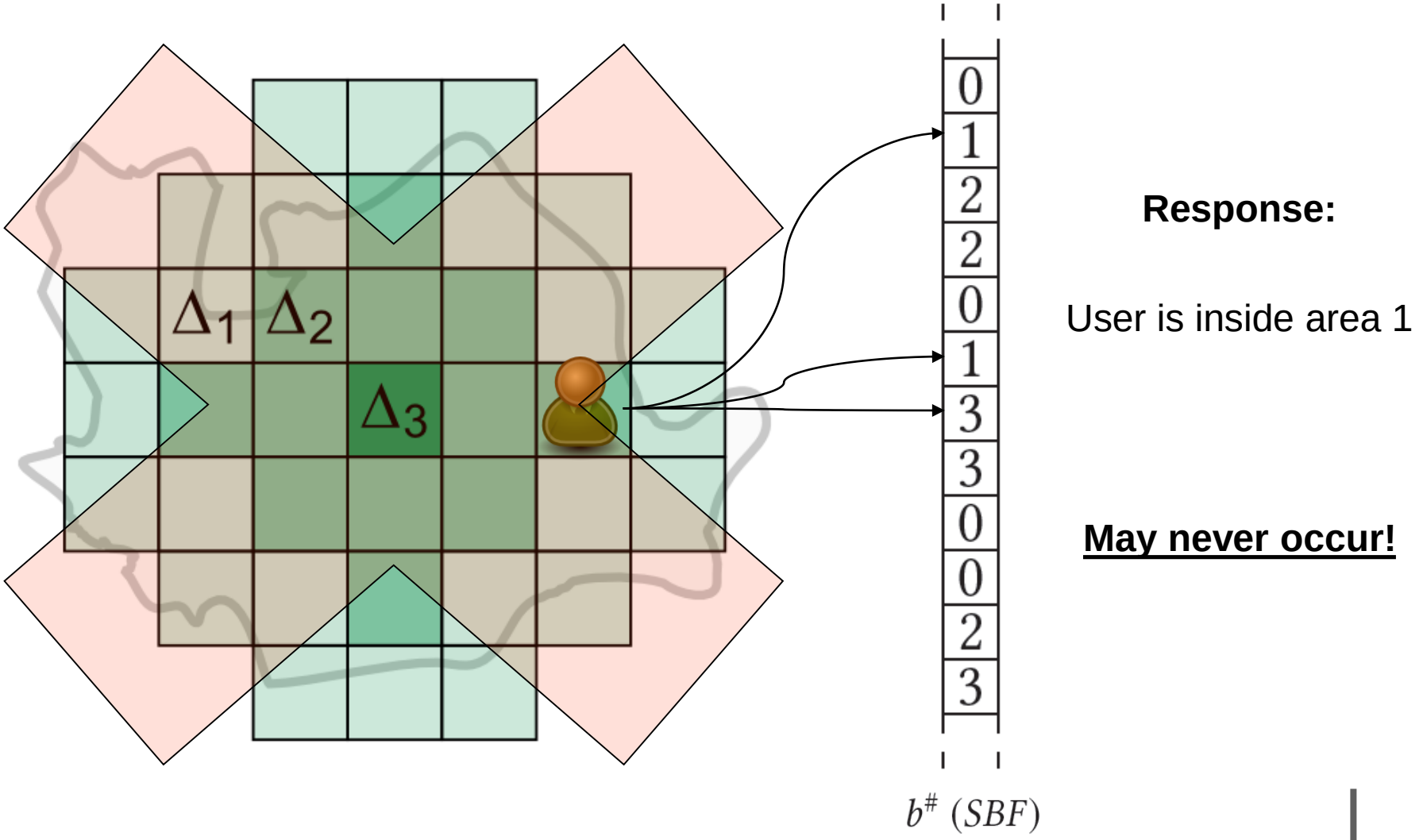
Example: inter-set error (1)



Response:

User is inside area 3

Example: inter-set error (2)



Response:

User is inside area 1

May never occur!

SBF properties

- SBF do not admit false negatives.

  ➔ There is no chanche for a person inside one of the monitored areas not to be noticed and to be deemed to stay outside all the AoI.

- *Fpp* and *isep* may be expressed as a function of *m*, *n* and *k.*

  ➔ Filters may be tuned in order to limit false positives and inter-set errors depending on the application context.

- AoI featuring a greater label are less exposed to false positives and inter-set errors. Moreover, area *j* can never be exchanged with area *i*, when $i < j$. Again, inter-set errors are not possible at all on the last AoI.

  ➔Thus, SBF supports a native priority regulation for areas of interest, a property which can be especially useful for security applications.

Homomorphic encryption

- In order to preserve both user's and provider's privacy we need one more element: some form of *homomorphic encryption*.
- We apply the multiplicative scheme provided by the *Paillier Cryptosystem*, an asymmetric encryption scheme featuring several homomorphic properties.

$$\forall m \in \mathbb{Z}_n, k \in \mathbb{N}: \quad D(E(m)^k \bmod n^2) = km \bmod n$$

- This way, we are able to store the encrypted version of the SBF on each client. The provider side, which holds the Paillier secure key, will decrypt the result of a proper operation (Private Hadamard Product) performed by each client before sending his position.
- The decrypted information allows the provider to understand (only) the area in which the user is supposed to be.

Security definition

In the two-party setting implemented by the protocol proposed in the following, the computation is achieved privately if at the end of the protocol execution:

the provider learns only *i in {1, …, s}*

and

the user learns nothing.

Information distribution

USER		*PROVIDER*	
$\mathcal{E}$	*(conventional grid)*	$\mathcal{E}$	*(conventional grid)*
$Enc(b^{\#})$	*(encrypted SBF)*	$\Delta_1, \ldots, \Delta_s$	*(AoIs)*
$h_1, \ldots, h_k$	*(hash set)*	$b^{\#}$	*(SBF)*
$hs_1, \ldots, hs_k$	*(hash salt)*	$h_1, \ldots, h_k$	*(hash set)*
		$hs_1, \ldots, hs_k$	*(hash salt)*
		$Pk^{\#}, Sk^{\#}$	*(homomorphic key pair)*

Protocol execution (1)

Before the communication starts …

1. First of all, the provider selects the *AoI*s $\Delta_1, \ldots, \Delta_s$
2. Then, he selects the desired false positive probability *fpp*, and determines *k* and *m* accordingly.
3. The provider choses *k* hash functions and produces an hash salt for each of them.
4. The provider computes the *SBF* $b^{\#}$ over *S*.
5. The service provider generates a public and private key pair ($Pk^{\#}$, $Sk^{\#}$) using a multiplicative homomorphic encryption scheme.
6. The SBF is encrypted with the public key in order to conceal *AoI*s.

Protocol execution (2)

When the communication starts …

1. The service provider sends the user the following information:
   - The conventional grid
   - The set of hash functions
   - The hash salt
   - The encrypted SBF (*Enc($b^{\#}$)*)

Protocol execution (3)

At regular time intervals or when required by the specific application …

1. The user determines his geographic position and selects the corresponding element e_u in the conventional grid.
2. Then, using the same hash set, the user builds the SBF over the sole element e_u ($b_u^{\#}$).
3. The user counts the number z of non-zero cells in $b_u^{\#}$.
4. The user computes the private Hadamard product $e^{\#} = Enc(b^{\#}) \bullet b_u^{\#}$ and applies a random permutation over $e^{\#}$.
5. The user sends $shuffle(e^{\#})$ and z to the provider.

Protocol execution (4)

On the provider side …

1. The provider side decrypts *shuffle*($e^{\#}$) and determines the number of non-zero values.
2. If the resulting number is < z, the user is outside any of the monitored areas.
3. Otherwise, the user is deemed to be inside the area Δ_i, where i is the smallest non-zero value among those returned.

Security analysis (1)

- The user never learns which areas are being monitored as AoIs are mapped in the encrypted SBF.
- The service provider never learns the exact position of the user, as he only learns the area in which the user is, subject to a false positive probability.
- If the user stay outside any of the monitored areas, the provider learns nothing.

Security analysis (2)

- The provider side may study the non-zero values returned by the user in order to infer the user position from these patterns.
- An exhaustive search on all the possible positions on the grid may reveal the user position even when he is outside the areas of interest.
- Actually, each pattern may be produced by a number of elements in the conventional grid. Hence, even in this case, the user position is concealed by *a-anonimity*.
- Assuming a linear distribution of values from *1* to *s* inside the filter, if $w \leq z \leq k$ is the number of non-zero values returned, the average value of *a* can be determined as

$$\bar{a} = \frac{|\mathcal{E}|}{\sum_{w=1}^{k} \binom{s+w-1}{w} + 1}$$

Comparative assessment

- In 2016 Solomon et al. proposed a comparative assessment between three among the most promising techniques aimed at preserving location privacy.

Comparison of privacy guarantees.

Method	User location privacy	DO privacy	Query privacy
SBF	*k*-anonymity based on filter size. DO learns when user is in an AOI	User only knows if location overlaps AOI	DO only sees obfuscated results - cannot correlate query to user
SkNN	DO only learns when user location overlaps an AOI	User only knows if location overlaps AOI	Extended protocol: DO/SP cannot correlate queries to users
HCT	User location is never shared with SP, DO, or any other user	*k*-anonymity guarantees based on fan-out (F) value	Limited, SP learns proximity of user to AOI with $1/F$ accuracy

Factors affecting performances of a single location query.

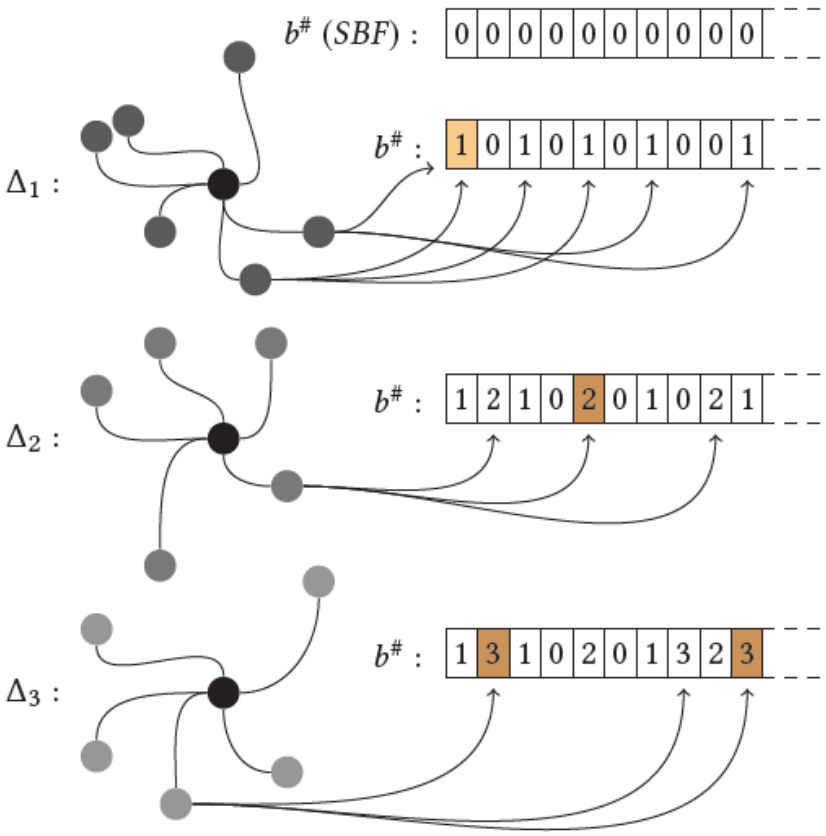
Method	Num AOIs	AOI Size	Grid Size
Spatial Bloom Filter (SBF)	No	No	No
Secure k-Nearest Neighbor (SkNN)	Yes	No	No
Hilbert Curve Transformation (HCT)	Yes	Yes	Yes

SBF: inter-network routing (1)

Our goal: to obtain anonymous routing between different neworks.

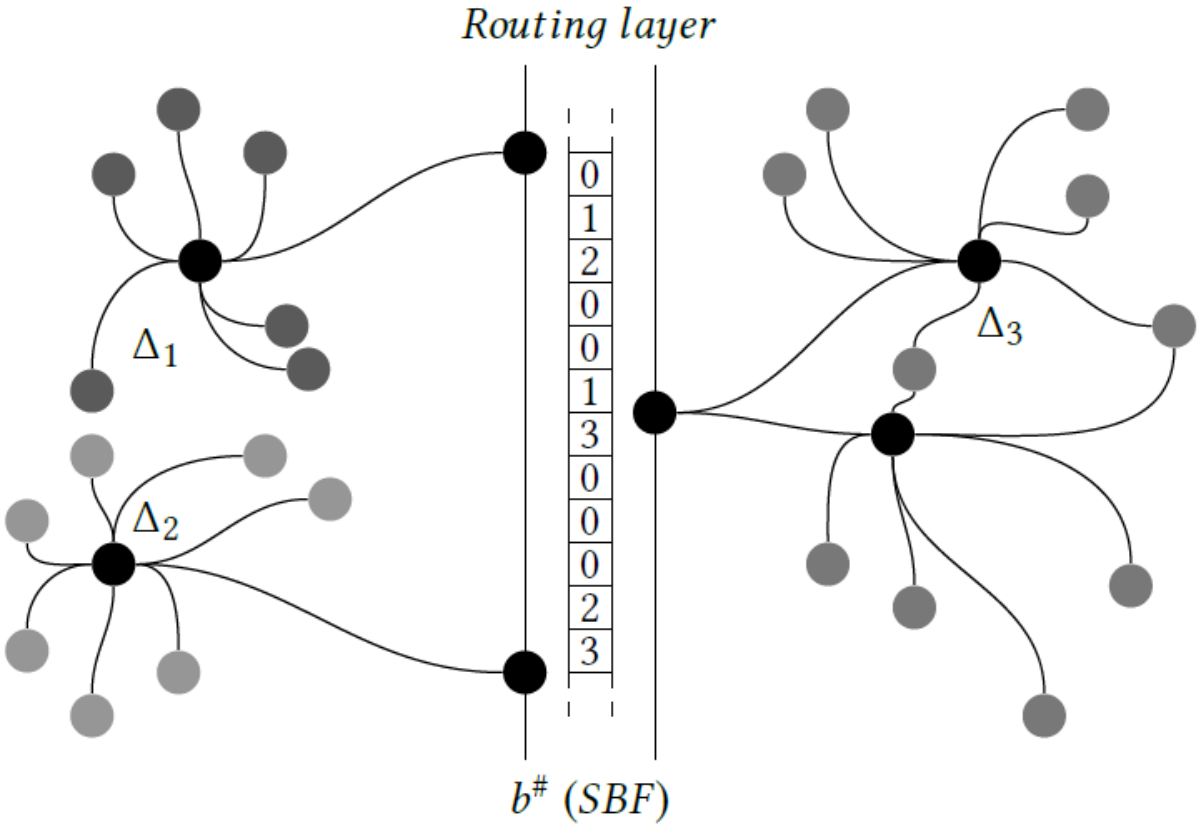
The routing layer: An abstraction of the network infrasctucture that interconnects the different networks (and provides routing).

Each node is provided with a unique **ID** and each subnetwork is provided with a unique **label**. Thus, each subnetwork can be represented as a set of its node identifiers.



SBF: inter-network routing (2)

The routing layer should be able to route messages to the correct subnetwork without knowing the receiver IP or ID.



Conclusion (1)

- **Location privacy** represents one of the most important challenges we need to face.
- Several techniques designed to conceal the user's location while keeping the location-based service operational were proposed in literature.
- Among them, we discussed the approach based on **Spatial Bloom Filters**, a probabilistic data structure designed to support membership queries on several sets efficiently.
- SBF were originally designed to handle location-based information and **ensure both users and provider privacy**.
- To this purpose SBF was coupled with a form of **homomorphic encryption** relying on the **Paillier cryptosystem**.
- **SBF may be used in a number of different scenarios** apart from the location-based one. Inter-network routing or vehicle routing are just an example.

Conclusion (2)

Thank you!

We have implemented the SBF data structure both in C++ and Python.
All the code is free for use (LGPL) and can be reached here:

https://github.com/spatialbloomfilter

For more information:

sbf.csr.unibo.it

Luca Calderoni

Dept. of Computer Science and Engineering

luca.calderoni@unibo.it

www.unibo.it