Location Privacy Preserving Mechanisms

Friederike Groschupp
Betreuer: Sree Harsha Totakura
Seminar Future Internet SS2017
Lehrstuhl Netzarchitekturen und Netzdienste
Fakultät für Informatik, Technische Universität München
Email: friederike.groschupp@tum.de

ABSTRACT

Location privacy is not an issue every user thinks of when using Google Maps, a fitness tracker or his GPS navigation system. Yet location privacy is an important part of one's anonymity while using location-based services. If location privacy is compromised, an adversary can draw sensitive conclusions about the user or use the gained information for his advantage. In the recent years, different approaches have been developed aiming to provide location privacy for different use cases. This paper aims to give an overview over the functionality, features and drawbacks of several important approaches: cloaking, mix zones, dummy queries, peer-to-peer systems and private information retrieval.

Keywords

Location-Based Services, Location Privacy, Anonymity

1. INTRODUCTION

A Location-Based Service (LBS) is an application that uses geographical information provided by a user in order to offer a service [12]. Today, LBSs are widely spread and used e.g. for vehicle or pedestrian navigation, providing location-based information or receiving a service at a specified area.

LBSs have become more popular and are used more often, resulting in users disclosing their location information more often. It is important to think about the user's information security and the threat to his anonymity. With the location information disclosed, an adversary, that is suspected at the LBS, might be able to draw conclusions about a user's lifestyle. Solely the fact that it is known that one was at a specific location can be bothersome. Additionally, anonymous messages sent from a location can be matched to the sender if it is known that they were present at the time. If the user has revealed their position knowingly or unknowingly in a previous message and now sends a message that is supposed to be anonymous and contains the same location information, the two messages and therefore the identity can be linked.

This paper discusses several approaches to preserve location privacy. At first, the problem is described in section 2, together with the use cases of LBSs, the protection goals, the assumptions about the system model, the adversary and the threats. Section 3 briefly introduces k-anonymity and location servers, two basic concepts used for location privacy. The location privacy approaches itself are discussed in Section 4: Temporal and spatial cloaking (4.1), mix zones (4.2) and dummy queries (4.3), together with the security they provide and possible attacks against them. A brief overview over peer-to-peer systems and private information retrieval is given in sections 4.4 and 4.5, respectively.

2. PROBLEM STATEMENT

The goal of this paper is to present an overview of the state-of-the-art approaches to provide location privacy. Location privacy is defined as the capability of precluding other parties than the ones trusted from learning the user's current or former location [1].

The main concern while discussing the issue of location privacy is on the part of the LBS. It is assumed that a LBS logs the received service requests containing location information. As a LBS is seen as a non-trusted party, the goal of location privacy-preserving mechanisms is to prevent accumulation of identifiable location information.

2.1 Use-case Scenarios

In order to discuss approaches aiming to protect location privacy, it is important to understand in what scenarios users may request a service from a LBS. The use cases for LBSs are diverse; some of them are mentioned below. Note that this list is not exhaustive.

- **Retrieving location specific information:** Users often want to gather information about their surroundings, be it finding a good restaurant nearby or the closest hospital or retrieving the weather forecast. Many of those requests are nearest-neighbor or range queries [2]. The technology most used for obtaining the location information for this kind of queries is the Global Positioning System (GPS).
- **Route planning and traffic information:** The use of a GPS-based guidance system while traveling along routes has become a frequently used application. While route planning and guiding is used for journeys the user is unfamiliar with, services like current traffic updates or hazard warnings are also used for often frequented routes [11].
- **Place bound use of services:** Recently, applications whose service is triggered when the user enters a certain area are becoming more popular. An example of such applications is a reward system, which offers discounts when a user enters a certain shop [1]. While

doi: 10.2313/NET-2017-09-1_02

some of these applications are based on GPS information, pervasive computing tools can also be used.

As we see, the situations in which users reveal their location may have very distinct characteristics and user goals. Every situation brings different problems with it. Therefore, the mechanism used in order to preserve location privacy has to fit the requirements of each situation.

2.2 Protection Goals

It is assumed that with a single message containing location information L it is not possible to draw conclusions about the user. This means that the sender of the message cannot be identified due to L. Other information like a username or metadata carried by the messages sent from the user to the LBS may, depending on the application, identify the user. Yet this is not the concern of location privacy.

Golle et. al [5] show that user identity can be drawn from several disclosed locations. If an application is able to link several queries containing location information and some of the locations correspond to the user's home and/or workplace, they might be easy to identify. In those cases where the identity of a user is - willingly or unwillingly - revealed, it should not be possible to link subsequent location updates to the user.

2.3 Assumptions

The approaches are based on similar assumptions regarding the system model, the user and the adversary. These assumptions are presented below.

System Model

In the model we discuss, there is no means of locating the user other than by the provided location information. While other methods such as locating the user e.g. by their IP address are available, they either have been deemed inaccurate [6] or are made difficult through proxies or sufficient use of obfuscation. Furthermore, it is always assumed that the user's mobile device comes with a position sensor (i.e. GPS receiver) and the user provides their information willingly and knowingly.

The applications considered in this paper do not need the user's real identity for providing their service; they are able to work with pseudonyms or without any user-related information.

Adversary

The adversary we regard is the LBS, be it either that the service provider is hostile or its system has been compromised. The user's device is deemed trustworthy, meaning that no malicious software or attacker has access to the position information on the device.

The following assumptions about the adversary apply to the mechanisms presented [11]:

- **Adversary cannot access client's identifiers:** It is assumed that the adversary has no access to the client's network addresses, e.g. their IP address, or needs other client specific information, like a username. This information is sufficiently protected, e.g. through the use of an anonymizing network or through the use of proxies. He is only able to observe the location information provided in the service requests.
- **Adversary can be active or passive:** When an adversary is passive, it only observes the location information provided in the queries and tries to draw conclusions from it.

  In contrast, an active adversary may modify the content of the responses from the LBS in order to trigger a certain behavior of the client. With this change of behavior, it might be simpler to deduce which of the locations is the real one. An additional strategy of an active adversary is to spoof false user information into the system in order to tamper with the results of the anonymization process.
- **Adversary has statistical background knowledge:** The adversary may have access to statistical knowledge, e.g. traffic densities at different locations. This information can be used to infer the real location.
- **Prior knowledge:** If the adversary monitors a specific user, they might have prior knowledge about the user [6].

2.4 Threats

The main purpose of location privacy is to provide sender anonymity, meaning that the adversary is not able to determine the identity of the originator of a message. When using a LBS, the main problem is the location information that is provided by the user. The requester of a service may be identified by correlating the location information with prior knowledge or easily researched information, like someone's home or workplace address.

When a user S sends a Message M containing location information L to a LBS that is the adversary A, sender anonymity and location information are threatened in the following ways [6]:

- **Restricted space identification:** If A knows that only S can be at L (e.g. L is in the area of a residential home), A can conclude that S is in L and has sent M. A trivial search in telephone books or property listings can reveal S's real identity.
- **Observation identification:** If A knows S is positioned at L and observes a message sent from the area of L, A can infer that S may have sent the message. If the user has revealed their identity and location in a former message, a subsequent anonymous message can be linked to it via the location.
- **Location tracking:** If A knows that S was or is at location L_i and has a linked series of location updates $L_1, L_2, \ldots, L_i, \ldots, L_n$, A knows that S visited all of these n locations. S might not want some of the locations they visited to be linked to them.

3. BASIC CONCEPTS

Some of the approaches are based on the same underlying concepts, which are k-anonymity and a trusted third party. These two concepts are briefly introduced below.

k-anonymity

k-anonymity states that within a specific set, a user is indistinguishable from at least $k-1$ other users [9]. In other words, a set is k-anonymous if it includes the user and at least $k-1$ other users identical to it in regards of the attributes considered.

A user's location can be represented by a tuple containing several dimensions: $[x_1, x_2]$, $[y_1, y_2]$ and $[t_1, t_2]$ [6]. $[x_1, x_2]$ and $[y_1, y_2]$ describe the two dimensional area in which the user is positioned. This information is always necessary. In addition, the time period in which the user was located in the area can be determined by $[t_1, t_2]$. If $x_1 \neq x_2$, $y_1 \neq y_2$ and $t_1 \neq t_2$, the tuple does not give away the exact information of the user, but only a certain range. The tuple is k-anonymous when the area it describes encompasses the position of the user and at least $k-1$ other user.

When an approach relies on k-anonymity or anonymity sets it is important to note, that anonymity is only provided in regard to location information. Other service-specific information or prior knowledge of the adversary could still identify the user [6]. For example, when Alice and Bob form one anonymity set and the adversary knows their genders, the adversary can infer that a service request for a women's clinic most probably originates from Alice.

Location Server

Some of the privacy approaches assume the existence of a trusted third party (TTP). This TTP is often called a location server (LS). The users of an anonymizing protocol in an area subscribe to the LS. The task of an LS is to receive the location provided by the user and anonymize it according to the selected approach. It then sends the query with the processed information to the LBS on behalf of the user and receives the response [10]. As the response is tailored to provide the service for all possible points in the enlarged, anonymized area, the LS filters the response. It then forwards the accurate response to the user.

The use of a LS in order to achieve location privacy has been viewed critically. The LS is an other third party that is used, and the most difficult step would be to identify a trustworthy instance. Building an infrastructure of trusted LS would take a lot of effort. Additionally, the TTP is a single point of attack for an adversary. If compromised, the attacker gets hold of all data, uncensored [11].

An other issue is that algorithms based on a TTP rely on other users using the same LS. They need to be present in the closer area in order to anonymize properly and still be able to provide sufficient information. Even if the required number of users is present, the location information the LBS receives will never be accurate but always an enlarged area. This may entail an information overhead, as the server has to provide replies for all possible locations encompassed in the area.

But the use of a TTP also brings an advantage for the user with it: The computations needed for the anonymization process are not done by the user. The user is relieved from heavy computations and other algorithm related details.

4. LOCATION PRIVACY APPROACHES

4.1 Cloaking

The concept of spatial and temporal cloaking [6] uses the concept of k-anonymity and a TTP in form of a LS, as introduced in section 3. The idea of the cloaking approach is to construct a tuple with its comprised ranges (space and/or time) as small as possible that is still k-anonymous. With cloaking, the LBS will never receive the exact information but only a certain interval in which the true location information is included. Therefore, it is important to consider the requirements in order to meet a certain performance and quality of results. Gruteser and Grunwald [6] distinguish between different application areas based on their need on exact spatial or temporal information. Increasing the accuracy of one information attribute can be done by decreasing the accuracy of the other.

Spatial Cloaking

The idea of spatial cloaking is computing a so-called cloaked area that encompasses the user and at least $k-1$ other users. This cloaked area is then forwarded to the LBS. One option to implement spatial cloaking is with a quadtree-based algorithm [6]. The general assumption is that by decreasing the level of required accuracy, k-anonymity can be provided in every situation. The algorithm is provided with the user's information, the parameter k_{min}, which determines the minimum size of the anonymity set, the area covered by the anonymizer and the information of all other users in the area. The detailed algorithm is presented in Listing 1. In short, the algorithm quarters the area considered as long as there are at least $k-1$ other users in the same area as the user. The smallest quadrant that still fulfills this constraint is then returned.

Temporal Cloaking

If a more exact spatial information for a service is required, one can make use of temporal cloaking, where the temporal accuracy is reduced. As all users being present in the area in a certain time interval and not just at one point of time are considered, the number of users available for anonymization increases. For temporal cloaking the user request is delayed until k_{min} other users have resided in the area determined by a resolution parameter. The resolution parameter determines how inaccurate the location information is allowed to be. The time range $[t_1, t_2]$ is then computed as following: t_2 is the current time, t_1 the time of the user request minus a random cloaking factor. This random cloaking factor is important, as the original exact information could be derived from t_1 otherwise. The tuple containing spatial and temporal information is then returned.

Accuracy of Results

Due to the lack of appropriate real life data, simulated automotive traffic flow for different city areas based on released detailed transportation data was used to test the algorithm [6]. According to the results of this experiment, the accuracy achieved by the algorithm differs based the structural characteristics of the area the user is located in. Areas containing highways have a higher density of vehicles, the median resolution in these areas ranges from 30 to 65 meters with $k_{min} = 5$. For areas mainly comprised of less frequented collector streets, the median resolution decreases to 125 to 250 meters for $k_{min} = 5$. For all scenarios, the mean size of the anonymity set computed is approximately 10 users.

If the application requires a certain minimum resolution, temporal cloaking can be added. If, for example, the provided location has to be exact within 15 meters, a time interval of 30 seconds is required in average for highway areas. At least 70 seconds are normally required for collector street areas [6]. This leads to the conclusion that, as expected, resolution is negatively correlated to the anonymity constraint. Additionally, when the service is not critical in regards of time, the spatial resolution can be increased by decreasing the temporal resolution.

That the algorithm is based on quartering the area causes the mean anonymity to be approximately twice the anonymity constraint. This indicates that an improved algorithm with better discretization could yield a better resolution with a lower mean anonymity closer to the anonymity constraint. This could be achieved for example by dividing the area according to each situation or merging areas that do not fulfill the anonymity constraint on their own.

Security Analysis

One potential active attack to circumvent cloaking approaches is by reporting a large number of additional locations to the LS. This can for example be done by the adversary spoofing false requests to the LS. This results in the LS releasing very accurate location information, as the anonymity constraint is fulfilled for a smaller area. However, the LS can be protected by only accepting one location information from each authenticated user. Acquiring a large number of authentication keys/ authenticated users should be made disproportionately expensive for an adversary.

Problems may emerge when several users issue a request at the same time [6]. A critical situation is depicted in figure 1, where circles represent users positioned in areas described by coordinates $([x_1, x_2], [y_1, y_2]);\ x_1, x_2, y_1, y_2 \in 0, 1, 2$. Let us assume that vehicles 1 to 4 use the same LS at the same time for anonymization. Then, the following requests with overlapping location information is received by the LBS:

$$\begin{aligned}\text{vehicle 1: } &([0,1],[0,1])\\ \text{vehicle 2: } &([1,2],[0,1])\\ \text{vehicle 3: } &([0,1],[1,2])\\ \text{vehicle 4: } &([0,2],[0,2])\end{aligned}$$

The first three vehicles name their position with three adjacent quadrants. The fourth one however issues the request with a quadrant larger than the others that covers them. Now, if all position information was processed with $k_{min} = 3$ by the LS, the adversary can conclude that vehicle 4 issued the request from the quadrant $([1, 2], [1, 2])$. This is because the algorithm would have returned a smaller quadrant otherwise. The k-anonymity constraint can be violated by overlapping simultaneous requests.

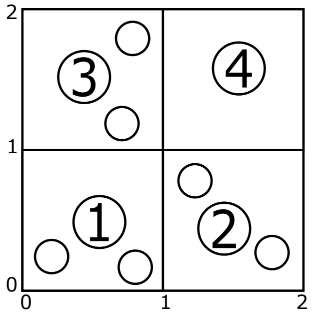

Figure 1: Compromised anonymity through several issued requests. Source: [6]

4.2 Mix zones

Mix zones are a useful approach when considering pervasive computing scenarios. For this scenario, the user is able to register with location-based applications for callbacks when a user enters a specific zone. The registration is done via a middleware that serves as proxy and as LS. A use case is to register with a service in order to receive advertisements and discounts when entering a shop. Most of these applications are not in need of the user's real identity and therefore able to work with short-term pseudonyms.

The idea of the approach presented by Beresford et. al [1], is to define areas called mix zones. For a group of users the mix zone is defined as a connected spatial region in which a user's position is not known. This is achieved by no user sending location updates. When a user enters a mix zone, all pseudonyms of users within that zone are changed. Within a mix zone, the user identities are "mixed", it cannot be distinguished between different users. Consequently, an observing attacker can not link users coming out of the mix zone with the ones going in.

The areas where users are registered for callbacks are called application zone.

Anonymity Sets

In order to quantify the anonymity provided, the concept of anonymity sets is used [1]. An anonymity set for a user u that visits a mix zone at time period t is the group of people that visit the mix zone during the same period t. The size of this anonymity set offers a first criteria for the anonymity provided. The more people are in it, the more anonymity is provided. The user may want to define a lower threshold, when fewer people are in the anonymity set they might refuse to provide their location in adjacent application zones. Additionally, when the average size of the anonymity sets of the mix zones surrounding an application zone is known, the expected anonymity level can be presented to a user before they sign up for a service provided in the application zone.

```
1. Initialize the quadrants $q$ and $q_{prev}$ as the total area covered by the anonymizer
2. Initialize a traffic vector v with the current positions of all known vehicles
3. Initialize $p$ as the position of the requestor vehicle
4. If number of vehicles in traffic vector v is less than $k_{min}$, return quadrant $q_{prev}$
5. Set $q_{prev}$ to $q$
6. Divide $q$ into quadrants of equal size
7. Set $q$ to the quadrant that includes $p$
8. Remove all vehicles outside $q$ from the traffic vector v
9. Repeat from Step 4
```

Listing 1: The spatial cloaking algorithm as presented in [6]. It computes an area containing the requesting user and at least $k-1$ other users.

Security Analysis

Until now, the model and its assumptions about the provided anonymity have presumed that the location of exit is independent to the point of exit. In reality, this is normally not the case [1]. Consider a mix zone with two entry points. When two users enter the mix zone at the two different entry points, they most likely will continue to walk in the same direction they entered the mix zone and exit the zone at the respective opposite site. Only in a very small number of cases, both will turn around and leave the zone through the same point that they entered it. Therefore, the user can be tracked with very high probability.

Entropy

A quantitative metric to measure the degree of anonymity is entropy. The entropy for a mix zone z can be computed based on recorded user movements. For each user traveling through z at time t, the preceding zone p, visited at $t-1$ and the subsequent zone s, visited at $t+1$, are noted. How often each possible pair (p,s) occurred, can be summarized in a movement Matrix M [1].

The probability with which the pair (p,s) occurs, can be calculated by dividing the number of (p,s) in the matrix through the sum of all pairs:

$$P(prec = p, subs = s) = \frac{M(p,s)}{\sum_{i,j} M(i,j)}$$

Ensuing, the conditional probability that the user exits through s, having entered through zone p, can be calculated as:

$$P(subs = s | prec = p) = \frac{M(p,s)}{\sum_{j} M(p,j)}$$

The information content associated with a number of possible outcomes with probability p_i can be calculated with Shannon's classic measure of entropy as:

$$h = -\sum_i p_i \cdot \log p_i$$

The lower the entropy is, the more certain an attacker can be about a true answer, and therefore the lower the degree of anonymity will be.

Beresford et. al [1] conducted experiments in their laboratory, where the movement of each staff was tracked with the help of position sensors. They also discuss the same situation as described before, where it is observed that two users enter the mix zones at opposite entrances, and two users leave the mix zones at the respective exit points. According to the data they gathered, the probability that both went straight in the observed situation is 99.9%. The possibility, that they did a U-turn, is 0.1%. The exact data for this results can be found in [1]. The entropy for the above observations can be calculated as

$$h = -(0.999 \cdot \log 0.999 + 0.001 \cdot \log 0.001) = 0.011$$

The entropy is much smaller than 1, therefore, an adversary is able to link pseudonyms and users identities with great success.

If movement possibilities are not equiprobable, an attacker with statistical background knowledge will be able to link user identities with high certainty and the concept of mix zones deems to provide only a low level of anonymity. Only if movement profiles are equally likely or a hostile observer has no statistical background knowledge, mix zones are able to provide anonymity. However, this still requires that the anonymity set is of sufficient size.

4.3 Dummy Queries

One solution to achieve location anonymity without the need of a TTP is the creation of dummy queries. The user does not only request the provided service for one, but for many locations [7]. For a basic dummy-based approach, the service request from the user to the server has the following format:

$$S = \langle u, L_1, L_2, \ldots, L_k \rangle$$

u is a user identifier (that does not permit possibilities to infer the real user identity), $\langle L_1, L_2, \ldots, L_k \rangle$ is a set of $k-1$ dummy locations and the real location. The LBS processes the request s and answers with the response R:

$$R = \langle (L_1, V_1), (L_2, V_2), \ldots, (L_k, V_k) \rangle$$

where V_i is the respective value for L_i requested from the service for $1 \leq i \leq k$. The user then only selects V_r, the value for their real location $L_r, 1 \leq r \leq k$. Note that only the user should know the value of r [7].

These queries have to be constructed carefully, as it would be easy to figure out the false queries when the adversary monitors the user over a longer period of time, uses statistical knowledge or checks the locations for validity (a location in the middle of a lake seems unlikely to be the real position in most cases). Therefore, many thoughts have been given on how to generate realistic dummies. One of the approaches is SybilQuery [11].

SybilQuery is an algorithm presented and proposed by Shankar et. al [11]. It is fully decentralized and autonomous, based on k-anonymity and aims to provide the creation of realistic dummy queries. In contrast to the approaches described before, it does consider the user traveling along predefined routes.

In the model presented a LBS is a database storing a set of tuples $< l, v >$, where l is a location and v is a value associated with l, like the current traffic condition or points of interest. The user queries the LBS periodically for the values associated with their current location while they move from a starting location to a destination. In order not to reveal their real current location, the users sends $k - 1$ additional requests to the LBS. Those queries are generated in accordance with the $k - 1$ paths that are created at the beginning of the trip and resemble the real path.

Dummy Query Generation

When the user uses SybilQuery for location privacy preserving, they enter their destination and the security parameter k. SybilQuery then generates $k - 1$ synthetic start and end points, with them $k-1$ paths and consequently generates the queries. The details of the process are described below [11].

At first, the endpoints are generated (note that this term is used both for source and destination). In order to generate paths that can not be detected as synthetic paths, a database containing regional traffic statistics is adduced. It is important to note that this is not real-time traffic information, but a statistic of former traffic trends. The endpoint generator uses this data in order to produce endpoints that share characteristics with the original source and destination. Characteristics considered are for example the surrounding traffic density or the likeliness of a location being an endpoint for a trip (a shopping mall would be much more likely than a highway intersection).

As a next step, the path generator uses the k start and end points, which includes the real source and destination, and produces k paths. A path is represented as a sequence of way points. For doing so, it uses a database of regional maps. The path generator and the endpoint generator are only needed once at the beginning of the trip.

After all k paths are constructed, the query generator is triggered with the user's current location. Then, it mimics the user's movement along the real path. Simply spoken, when a user has traveled a certain distance d and sends an other request to the LBS, the generator applies similar offsets to the synthetic paths and sends $k - 1$ queries with synthetic locations. However, these simulations are more realistic if for example current traffic conditions are taken into consideration (e.g. if there are traffic congestions along a synthetic path, a slower advance along this path should be simulated).

Extensions

Considering an adversary that may be passive or active and may have statistical background knowledge, the following improvements to the basic principles can make the system more robust towards attacks [11]:

- **Randomized path selection:** A user may not always use the shortest route to their destination. If all synthetic paths correspond to the shortest path between their respective endpoints, the real path is to be figured out easily. Therefore, multiple paths should be generated for each pair of endpoints, the one used for the dummy queries is then chosen with a probabilistic method.
- **Robustness towards active adversary:** An active adversary may report false information like a traffic congestion. A real user may take a detour based on that information. SybilQuery has to mimic this behavior for the synthetic routes.

  Additionally, one way to detect active adversaries is to query multiple LBSs. The responses of the adversary may differ and it can be detected by them.
- **Caching of endpoints:** If the adversary monitors the user over a longer period of time, paths that the user travels frequently (e.g. from their home to their workplace) are easy to identify as the real path. This is because the other parts are generated randomly and therefore do not appear as often. Likewise, when a user finishes a path and starts a new one shortly after, the second set of synthetic paths do no match the first set, when former destination and new source lay too far apart. Those attacks are handled by SybilQuery caching used paths and endpoints.
- **Path continuity:** Even though the generated paths are alike, it is possible for some paths to take longer than the others. If the user has reached their destination before the synthetic paths do and the system stops sending queries, the LBS can spot the synthetic paths. Therefore, SybilQuery continues to imitate movement along the synthetic paths until their destination is reached.
- **GPS sensor noise:** The location provided by the GPS system is normally imprecise. If the locations of the dummy queries are always exactly positioned on the road, the synthetic paths can be detected. Therefore, a random noise is added to the location information for each dummy query.

Security Analysis

The system turns out to be relatively robust to attackers guessing the real path. Shankar et. al [11] conducted a user study, where they presented k paths in the San Francisco Bay Area to volunteers. All volunteers had good knowledge of the area and had location information tools at their disposal. The volunteers had to figure out the real path for $k = 4$ and $k = 6$. The volunteers were able to figure out the real path with a probability of 0.26 for $k = 4$ and 0.19 for $k = 6$. These values are close to the expected values for random guessing (0.25 and 0.17 for $k = 4$ and $k = 6$ respectively). This leads to the conclusion that the synthetic paths generated by SybilQuery resemble real paths.

4.4 Peer-to-Peer Systems

In order to avoid the necessity of a LS, peer-to-peer systems for anonymous location-based queries have been developed. One of those approaches is MobiHide [4], which is based on k-anonymity. In this approach, the users authenticate with a certification server (which is only used for authentication, not for the anonymization process) and self-organize thereafter. As there is no central LS storing the location of each user, the information has to be stored distributed. The system is based on the Chord [13] distributed hash table architecture. As this method works on one dimension, the user positions are mapped from the two-dimensional space into one-dimensional space, for example with the Hilbert curve. The Hilbert space filling curve is a continuous fractal that can be used for this mapping. The area considered is filled with the Hilbert curve, the location of each user is then determined as the offset of their location along the curve. Each user holds the information about their location in the one dimensional space as well as several pointers towards specified other users. With this architecture, each user is able to determine n neighbored users around him.

When a user wants to request a service from a LBS, MobiHide determines k users including the requester who are consecutive neighbors in the constructed one-dimensional space. Which of the k possible sets is chosen is decided randomly. The user then computes the smallest rectangle that covers all k selected users. The service request containing the area of the rectangle is the sent by the user to the LBS which responds with the respective information. The user then filters the response based on his location for the information he is really looking for.

Security Analysis

MobiHide provides k-anonymity in the case that all users issue queries with the same probability [4]. If this is not the case, the approach is vulnerable to the correlation attack. Consider the extreme case when only one user sends service request. Consequently, the adversary receives multiple rectangles of which it knows that it covers the user's location. The adversary can now intersect the rectangles in order to shrink the number of possibilities for the user's location. However, it is believed that the difference of number of requests per user is not as extreme, even when distribution is not expected to be perfectly even.

4.5 Private Information Retrieval

Private Information Retrieval (PIR) is used when a database is to be queried and the entity holding the database is not to know which entry was queried. One way to achieve this goal would be to present the whole content of the database to the requester and they read the entry they need. Yet this solution is not feasible for reality, as an immense overhead of data traffic is produced and the owner of the database may not want to give away their entire data.

Consequently, computational PIR [8] based on cryptographic assumptions have been developed. Ghinitha et. al [3] propose an algorithm based on the Quadratic Residuosity Assumption. The protocol imposes an overhead of $O(n)$ at the server and costs of $O(\sqrt{n})$ for client-server communication.

A problem that arises for the deployment of PIR schemes is that code modifications at server and client side are required. Therefore, PIR cannot be easily used for existing applications.

Security analysis

In contrast to the other approaches introduced, PIR offers strong cryptographic guarantees on anonymity. It also has significant advantages in regards to the information that is revealed to the LBS [3], as no kind of location information is disclosed. Other approaches submit an inaccurate location or several possible locations. PIR does not reveal any information at all. This is also a protection against correlation attacks.

No information at all is disclosed, this leads to a reduction of the identification probability. Let U be the number of all possible users, e.g. all mobile users in the country. The identification probability for PIR is $\frac{1}{U}$. For the approaches mentioned above, the identification probability is $\frac{1}{k}$, where k denotes the size of the anonymity set or the number of users of the service in the area. Normally, U should be significantly larger than k, implying that the identification probability is significantly lower.

5. CONCLUSION

Several approaches that aim to provide location privacy while using a Location-Based Service have been presented in this paper. The approaches differ in terms of the use cases they are appropriate for, the basic concepts they rely on, the overhead they produce, the accuracy they are able to provide and the attacks they are vulnerable to.

Mix zones are a good choice when working with predefined application areas, for example in pervasive computing. Cloaking methods are appropriate when a decrease of temporal resolution can be tolerated if it increases spatial resolution. Peer-to-peer systems have the advantage of being decentralized and abolishing the need of a TTP. A scheme like SybilQuery that produces dummy queries is convenient to use while traveling along paths and provides independence from other users. Private Information Retrieval is based on cryptographic guarantees and provides the strongest anonymity while producing the most overhead.

Of the location privacy-preserving mechanism presented, none is suited to provide location privacy in every possible use case. For each use case the requirements of the user and the characteristics of the application have to be considered.

For the future it is essential to raise the user's attention towards the importance of location privacy and the dangers when revealing ones location to third parties. In order to be successful, solutions have to be convenient to deploy and easy to use.

6. REFERENCES

[1] A. R. Beresford and F. Stajano. Location privacy in pervasive computing. *IEEE Pervasive Computing*, 2(1):46–55, 2003.

[2] S. Dhar and U. Varshney. Challenges and business models for mobile location-based services and advertising. *Communications of the ACM*, 54(5):121–128, 2011.

[3] G. Ghinita, P. Kalnis, A. Khoshgozaran, C. Shahabi, and K.-L. Tan. Private queries in location based services: Anonymizers are not necessary. *Proceedings*

of the ACM SIGMOD International Conference on Management of Data, 2009.

[4] G. Ghinita, P. Kalnis, and S. Skiadopoulos. Mobihide: A mobilea peer-to-peer system for anonymous location-based queries. In D. Papadias, D. Zhang, and G. Kollios, editors, *Advances in spatial and temporal databases*, volume 4605 of *Lecture Notes in Computer Science*, pages 221–238. Springer, Berlin, 2007.

[5] P. Golle and K. Partridge. On the anonymity of home/work location pairs. In H. Tokuda, M. Beigl, A. Friday, A. J. B. Brush, and Y. Tobe, editors, *Pervasive computing*, volume 5538 of *Lecture Notes in Computer Science*, pages 390–397. Springer, Berlin, 2009.

[6] M. Gruteser and D. Grunwald. Anonymous usage of location-based services through spatial and temporal cloaking. In D. Siewiorek, M. Baker, and R. T. Morris, editors, *Proceedings of the 1st international conference on Mobile systems, applications and services - MobiSys '03*, pages 31–42, New York, New York, USA, 2003. ACM Press.

[7] C. S. Jensen, H. Lu, and M. L. Yiu. Location privacy techniques in client-server architectures. In C. Bettini, S. Jajodia, P. Samarati, and X. S. Wang, editors, *Privacy in Location-Based Applications: Research Issues and Emerging Trends*, pages 31–58. Springer, Berlin, 2009.

[8] E. Kushilevitz and R. Ostrovsky. Replication is not needed: single database, computationally-private information retrieval. In *Proceedings of the 38th Annual Symposium on Foundations of Computer Science*, pages 364–373, Los Alamitos, Calif., 1997. IEEE Computer Soc. Press.

[9] X. Lu and M. H. Au. Chapter 11 - an introduction to various privacy models. In M. H. Au and K.-K. R. Choo, editors, *Mobile Security and Privacy*. Syngress, Boston, 2017.

[10] G. Myles, A. Friday, and N. Davies. Preserving privacy in environments with location-based applications. *IEEE Pervasive Computing*, 2(1):56–64, 2003.

[11] P. Shankar, V. Ganapathy, and L. Iftode. Privately querying location-based services with sybilquery. In A. A. Helal, editor, *UbiComp '09*, page 31, New York, N.Y, 2009. Association for Computing Machinery.

[12] S. Spiekermann. General aspects of location based services. In J. H. Schiller and A. Voisard, editors, *Location-based services*, Morgan Kaufmann series in data management systems. Morgan Kaufmann Publishers, San Francisco, CA, 2004.

[13] I. Stoica, R. Morris, D. Liben-Nowell, D. R. Karger, M. F. Kaashoek, F. Dabek, and H. Balakrishnan. Chord: A scalable peer-to-peer lookup protocol for internet applications. *IEEE/ACM Transactions on Networking*, 11(1):17–32, 2003.