Providing K–Anonymity in Location Based Services

Aris Gkoulalas–Divanis
Department of Biomedical
Informatics
Vanderbilt University
Nashville, TN, USA.
arisgd@acm.org

Panos Kalnis
Computer Science Dept.
King Abdullah Univ. of
Science & Technology
Jeddah, Saudi Arabia.
panos.kalnis@kaust.edu.sa

Vassilios S. Verykios
Computer & Communication
Engineering Department
University of Thessaly
Volos, Greece.
verykios@inf.uth.gr

ABSTRACT

The offering of anonymity in relational databases has attracted a great deal of attention in the database community during the last decade [4]. Among the different solution approaches that have been proposed to tackle this problem, $\mathcal{K}$–anonymity has received increased attention and has been extensively studied in various forms. New forms of data that come into existence, like location data capturing user movement, pave the way for the offering of cutting edge services such as the prevailing Location Based Services (LBSs). Given that these services assume an in–depth knowledge of the mobile users' whereabouts it is certain that the assumed knowledge may breach the privacy of the users. Thus, concrete approaches are necessary to preserve the anonymity of the mobile users when requesting LBSs.

In this work, we survey recent advancements for the offering of $\mathcal{K}$–anonymity in LBSs. Most of the approaches that have been proposed heavily depend on a trusted server component – that acts as an intermediate between the end user and the service provider – to preserve the anonymity of the former entity. Existing approaches are partitioned in three categories: (a) historical $\mathcal{K}$–anonymity, (b) location $\mathcal{K}$–anonymity, and (c) trajectory $\mathcal{K}$–anonymity. In each of these categories we present some of the most prevalent methodologies that have been proposed and highlight their operation.

1. INTRODUCTION

The enormous advances in positioning technologies like GPS, GSM, UMTS and RFID, along with the rapid developments in the wireless communications industry, have made possible the accurate tracking of user location at a low cost [3]. The increased tracking accuracy gave rise to a novel class of applications which are based on user location, spanning from emergency response and "search & rescue" services (such as E–911), to services that automate everyday tasks, such as online user navigation to avoid traffic jams and/or bad weather conditions, way–finding, store–finding and friend–finding, as well as mobile commerce and surveying. All these services, including localized news and state–of–the–art location–based games that merge physical and virtual spaces, require an extensive use of location data and are collectively known as *Location Based Services* (LBSs).

The benefit of LBSs both to the individual subscribers and to the community, as a whole, is undeniable. With respect to the public welfare, the collection of location data by a governmental or other public agency may enhance the process of decision making regarding tasks such as urban planning, routing, wildlife rescuing and environmental pollution. As is evident, the new computing paradigm is changing the way people live and work. However, it also poses a series of challenges as it touches upon delicate privacy issues [1].

1.1 Privacy challenges in LBSs

The offering of LBSs requires an in–depth knowledge of the subscribers' whereabouts. Thus, with untrustworthy service providers the deployment of LBSs may breach the privacy of the mobile users. Consider, for example, a service request originating from the house of a user. The request contains sufficient information to identify the requester, even if it lacks of any other identification data (e.g., the user ID, the user name, etc.). This is true since the mapping of the exact coordinates that are part of the user request to a publicly available data source of geocoding information can reveal that the request originated from a house and thus increase the confidence of the service provider that the requester is a member of the household. Moreover, if a series of requests for LBSs are matched to the same individual then it is possible for the service provider to identify places that this user frequently visits, reveal his/her personal habits, political/religious affiliations or alternative lifestyles, as well as build a complete profile of the user based on the history of his/her movement in the system. Consequently, without the existence of strict safeguards, the deployment of LBSs and the sharing of location information may easily lead the way to an abuse scenario, similar to Orwell's Big Brother society. To avoid this situation and adequately protect the privacy of the users when requesting LBSs, sophisticated algorithms have to be devised.

1.2 Organization of the rest of the paper

The rest of this work is organized as follows. In Section 2 we present the centralized model for the offering of privacy in LBSs along with an example of its operation. Section 3 highlights the working assumptions that are used by most of the state–of–the–art privacy preserving approaches in LBSs. Following that, in Section 4 we present a taxonomy of the existing centralized $\mathcal{K}$–anonymization approaches for the offering of privacy in LBSs. Finally, Section 5 concludes this work.

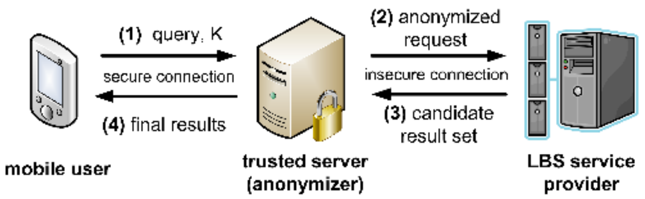


Figure 1: The centralized model for privacy in LBSs.

2. THE PRIVACY MODEL

Figure 1 presents a big picture of the centralized model for privacy in LBSs. In this model, we consider a population of users who are supported by some telecommunication infrastructure, owned by a telecom operator. Every user in the system has a mobile device that periodically transmits a location update to some traffic monitoring system residing in a trusted server of the telecom operator. The communicated location update contains the current location (and time) of the user and is stored by the trusted server. A set of LBSs are available to the subscribed users through service providers that collaborate with the telecom operator. We assume that these service providers are untrusted; if a user submits a request for an LBS directly to the service provider then his/her identity can be revealed and his/her privacy can be compromised. Motivated by this fact, the centralized privacy model requires that every user request for an LBS has to be submitted to a trusted server of the telecom operator via a secure communication channel. The role of the trusted server (anonymizer) is to filter the incoming user requests and to produce anonymous counterparts that can be safely forwarded to the service providers in order to be serviced. To produce the anonymous counterpart to an original user request, the trusted server has to incorporate algorithms that (a) remove any obvious identifiers that are part of the user request (e.g., ID, name) and (b) effectively transform the exact location of request into a spatiotemporal area (a.k.a. the *area of anonymity*) that includes a sufficient number of nearby users registered to the system so as to prevent the attacker from locating the requester. These users formulate the *anonymity set* of the requester.

2.1 An example of operation

The operation of the centralized privacy model is exemplified in Figure 2, where we assume a user Bob who asks for the nearest betting office B_i to his current location. This is a typical nearest neighbor query that is commonly met in LBSs. Bob forwards his query (request) Q to the anonymizer. Then the anonymizer, who has knowledge of the current location of each user in the system, identifies 3 users who are near Bob and encloses all four users in a region R. Subsequently, instead of sending Bob's location to the LBS provider, the anonymizer sends region R. When the LBS service provider receives R it computes all the betting offices that can be the nearest neighbor of any point in R. It is important to notice that although the service provider is certain that Bob is located within R, it has no means to identify the exact location of Bob in R. Using its database, the service provider generates a candidate set of answers (i.e. $\{B_1, B_2, B_3, B_4\}$) and forwards it to the anonymizer. The anonymizer uses the actual location of Bob inside R to filter out all the false hits and forward the actual nearest neighbor (in this case B_1) to Bob. This step concludes the provision of the LBS in a privacy aware manner. The way that region R is formulated, as well as the privacy guarantees that are offered to the requester by the system, are based on the specific privacy methodology that is employed by the trusted server.

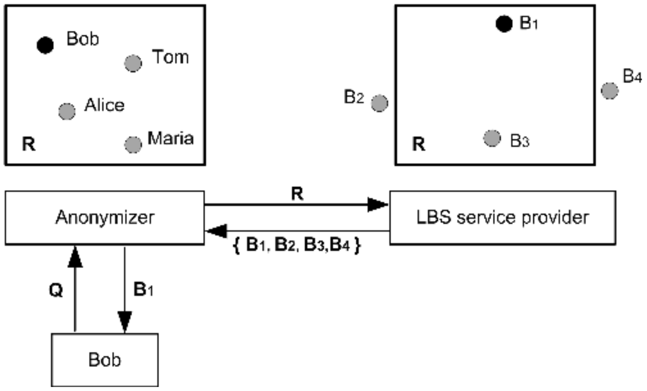


Figure 2: A use–case scenario of the centralized model.

3. WORKING ASSUMPTIONS AND THREAT MODEL

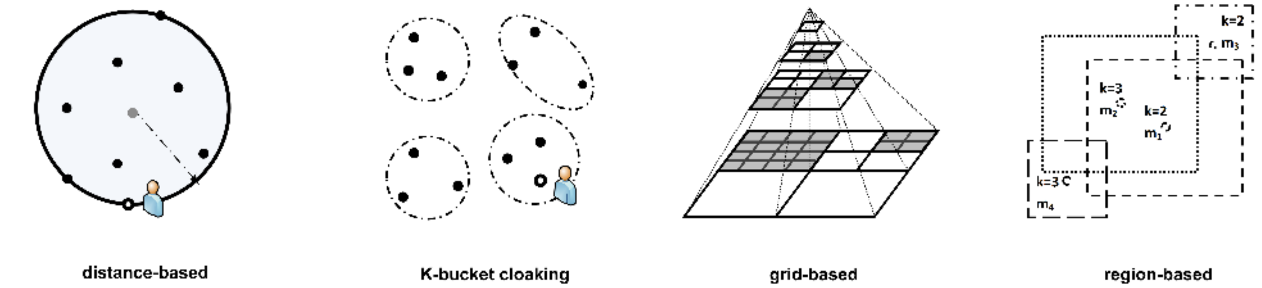
Several centralized $\mathcal{K}$–anonymity approaches have been proposed for the offering of privacy in LBSs. In what follows, we present the working assumptions about the capabilities of the attacker that are employed by most of these approaches. The knowledge of the assumptions is necessary to compare the different approaches in terms of privacy guarantees that they offer to the requesters of LBSs. Generally, the attacker is assumed to have the following capabilities:

1. The attacker can intercept the region where anonymity is offered to the requester of an LBS. This implies that the LBS service provider is untrustworthy.

2. The attacker has knowledge of the algorithms that are used by the trusted server to offer privacy in LBSs. This situation is common in the security literature where algorithms are typically publicly available.

3. The attacker can obtain the current location of all the users in the system. This assumption is motivated by the fact that users may often issue queries from easily identifiable locations. Since it is difficult to model the exact amount of knowledge that an attacker may have at his/her disposal, this assumption dictates that the privacy methodology must be provably secure under the worst–case scenario.

4. The attacker tries to breach the location privacy of the users by using only current location data; he/she is unaware of any historical information about the movement of the users, as well as any behavior patterns of particular clients (e.g., a user is often asking a particular query at a certain location or time).



(a) Data–dependent cloaking.

(b) Space–dependent cloaking.

Figure 3: Cloaking strategies for the offering of $\mathcal{K}-$anonymity in LBSs.

4. TAXONOMY OF EXISTING K–ANONYMIZATION APPROACHES

The main body of research for the offering of privacy in LBSs includes approaches that are based on the notion of $\mathcal{K}$–anonymity. $\mathcal{K}$–anonymity, originally proposed by Samarati and Sweeney [11, 12] in the context of relational data, requires that "each data release must be such that every combination of values of private data can be indistinctly matched to at least $\mathcal{K}$ individuals". In this sense, $\mathcal{K}$–anonymity requires that every record in a released dataset is indistinguishable from at least $\mathcal{K}$–1 other records with respect to a certain set of identifying variables. In the context of LBSs, the identifying variable is the location of the individuals when requesting LBSs; releasing a request of an individual for an LBS to an untrusted third party should make certain that the actual location of request cannot be associated (at least with a high probability) with the identity of the requester. To satisfy $\mathcal{K}$–anonymity in LBSs, the most widely adopted anonymization strategy is *cloaking*. In cloaking, the actual location of request is transformed into a bounded area that is large enough to contain the requester along with (at least) $\mathcal{K}$–1 other users. Cloaking ensures that the identity of the requester cannot be disclosed with a probability that is significantly larger than $1/\mathcal{K}$, among $\mathcal{K}$–1 other users. Some of the most prevalent cloaking strategies that generalize the actual locations of request to spatially bounded areas, are presented in Figure 3. They can be partitioned into two groups: *data–dependent cloaking* and *space–dependent cloaking* methodologies.

4.1 Data–dependent cloaking

Data–dependent cloaking strategies formulate the region of anonymity based on the actual location of each user in the system and his/her distance from the location of request. Specifically, distance–based cloaking algorithms (e.g., [1, 7, 15]) retrieve the $\mathcal{K}$–1 nearest neighbors of the requester and generate a region that includes all the $\mathcal{K}$ users. In $\mathcal{K}$–bucket cloaking (e.g., [2, 9]) the users are arranged into groups of $\mathcal{K}$ and the anonymity region is computed as the *Minimum Bounding Rectangle* (MBR) that contains the $\mathcal{K}$ users in the group of the requester.

4.2 Space–dependent cloaking

Space–dependent cloaking strategies take into consideration the total area that is covered by the anonymizer to formulate the regions of anonymity. Specifically, grid–based cloaking strategies (e.g., [6, 8, 10]) partition the area in a grid fashion and generate the region of anonymity by retrieving the users in each cell of the grid (starting from the cell of the requester and moving to neighboring cells) until at least $\mathcal{K}$ users are found. On the other hand, region–based cloaking strategies, such as [5], use the spatial properties of the area to generate rectangles centered at the location of request and to utilize them for the offering of $\mathcal{K}$–anonymity.

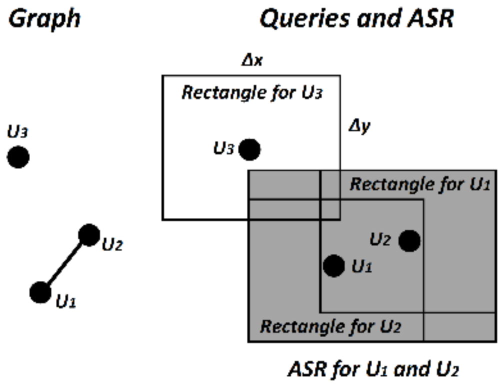


Figure 4: An example of *Clique Cloak*.

In what follows, we present the different research directions for the offering of $\mathcal{K}$–anonymity in LBSs. Alongside the presented methodologies, we include some examples to demonstrate their operation.

4.3 Location K–anonymity

Location $\mathcal{K}$–anonymity approaches (e.g., [5, 8–10]) protect user privacy by utilizing the current location (instead of the history of collected locations) of each user in the system. They operate on LBSs that require a single location transmission from the requesting party in order to be successfully provided (e.g., store–finder, friend–finder, etc.), instead of the communication of multiple location updates. The different cloaking strategies that have been proposed for the offering of location $\mathcal{K}$–anonymity are presented in Figure 3. In what follows, we detail over some of the most popular approaches in this category.

Clique Cloak [5] is a graph–based (region–based) approach that mutually anonymizes multiple incoming requests for LBSs. For each query that is received for servicing, the algorithm generates a rectangle centered at the location of the requester, with its sides being parallel to the considered

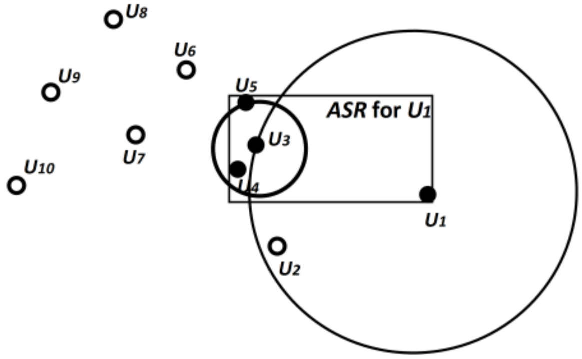


Figure 5: An example of *NN–Cloak*.

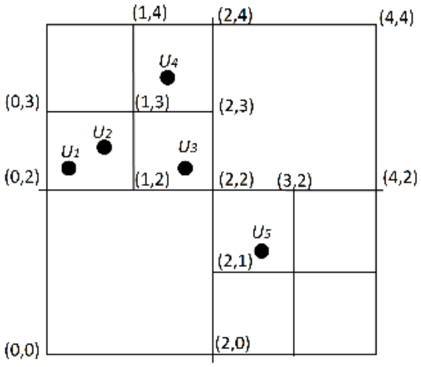


Figure 6: An example of *Casper*.

x and y–axis, respectively having Δx and Δy extents. The new query is then marked as a node in a graph for as long as it awaits for its anonymization. Two vertices (queries) in the graph are connected together if the corresponding users fall in the rectangles of each other. An edge of the graph demonstrates that the requester of each of the two queries can be included in the computed anonymity set of the other, and thus, a $\mathcal{K}$–clique of the graph shows that all the corresponding $\mathcal{K}$ requests can be anonymized together (thus offering $\mathcal{K}$–anonymity to all the $\mathcal{K}$ users that participate in the $\mathcal{K}$–clique). Finally, tight to each request is a temporal interval Δt that defines the maximum amount of time that this request can be retained by the system for its anonymization. If a $\mathcal{K}$–clique cannot be found within Δt then the request is dropped as unserviceable. Figure 4 demonstrates the operation of *Clique Cloak* in the case scenario where three queries for LBSs (located at U_1, U_2 and U_3) have been synchronously submitted to the trusted server. Assuming that $\mathcal{K} = 2$, the generated rectangles for U_1 and U_2 fall in each other and thus they form a 2–clique in the graph. As a result, the MBR enclosure of the respective rectangles (shown here in gray) represents the *Anonymity Spatial Region* (ASR) where 2–anonymity is offered to these users. On the other hand, the request of U_3 has to wait in the system until a new query (formulating a 2–clique with U_3) arrives. As one can observe, *Clique Cloak* may affect the quality of service that is offered to the users as the servicing of some queries may be substantially delayed, while other queries may be dropped as unserviceable. The approaches that follow do not suffer from these shortcomings.

Center Cloak [9] is a distance–based approach that provides a naïve solution to $\mathcal{K}$–anonymity in LBSs. In *Center Cloak*, the $\mathcal{K}$–1 nearest neighbors of the requester are retrieved and the ASR is computed as the MBR enclosure of all the $\mathcal{K}$ users. By construction, *Center Cloak* suffers from what is known as the "center–of–ASR" attack; the identity of the requester can be accurately guessed with a probability that far exceeds $1/\mathcal{K}$ as he/she is expected to be close to the center of the ASR. The "center–of–ASR" attack is an instance of a more general problem that is worth mentioning. Since cloaking algorithms are expected to be publicly available (see assumption 2 of Section 3), attackers can easily exploit any implementation decisions with respect to the placement of the requester to the generated ASRs. As an effect, several of the currently available approaches suffer from similar kinds of attacks.

A randomized variant of *Center Cloak*, which offers increased uncertainty regarding the location of the requester in the generated ASR, is Nearest Neighbor Cloak (*NN–Cloak*) [9]. In *NN–Cloak* the ASR is formulated as follows: Given a user query for an LBS, *NN–Cloak* first retrieves the $\mathcal{K}$–1 nearest neighbors of the requester. Second, it randomly selects one among the $\mathcal{K}$ users and identifies his/her $\mathcal{K}$–1 nearest neighbors. Finally, the $\mathcal{K}$–ASR is constructed as the MBR enclosure of the second set of $\mathcal{K}$ users, augmented (if necessary) to include the requester. Figure 5 presents an example where 3–anonymity is offered to user U_1 by using *NN–Cloak*. First, U_1 formulates set $S_1 = \{U_1, U_2, U_3\}$ with his/her two nearest neighbors. Second, *NN–Cloak* randomly selects U_3 from S_1 and computes his/her two nearest neighbors in the system. This leads to set $S_2 = \{U_3, U_4, U_5\}$. Finally, the MBR of S_2 is augmented to include the requester U_1, leading to the 4–ASR for U_1 that is shown in Figure 5. Since the probability of selecting the requester from S_1 when formulating S_2 is at most $1/\mathcal{K}$ (due to random choice), *NN–Cloak* is not vulnerable to the "center–of–ASR" attack.

Casper [10] is one of the most popular grid–based approaches to location $\mathcal{K}$–anonymity. In *Casper* the entire area that is covered by the anonymizer is divided in a grid–fashion and organized in a pyramid data structure of layers that is similar to a Quad–tree [13] (see the pyramid structure in Figure 3(b)). The top layer of the pyramid contains the entire area, whereas the lowest level of the pyramid collects the finest–grained granularity of the partitioning. Each cell in the lowest level of the pyramid has a minimum size that corresponds to the anonymity resolution. When a new query for an LBS is received by the trusted server, *Casper* locates the lowest–level cell in the pyramid that contains the requester and examines if this cell also contains $\mathcal{K}$–1 other users. If the cell contains enough users then it becomes the $\mathcal{K}$–ASR. Otherwise, *Casper* searches the horizontal and the vertical neighbors of this cell to identify if the number of users in each of these cells, when combined with the number of users in the cell of the requester, suffice for the provision of location $\mathcal{K}$–anonymity. If this is true, then the corresponding union of cells becomes the $\mathcal{K}$–ASR. Else, *Casper* moves one level up in the pyramid to retrieve the parent (cell) of the cell of request and repeats the same process until the $\mathcal{K}$ users that will formulate the ASR are found. Figure 6 provides an example of this cloaking operation. Assuming a request coming from cell $\langle(0,2),(1,3)\rangle$ (where (0,2) are the lower–left and (1,3) the upper–right coordinates of the cell) with an anonymity

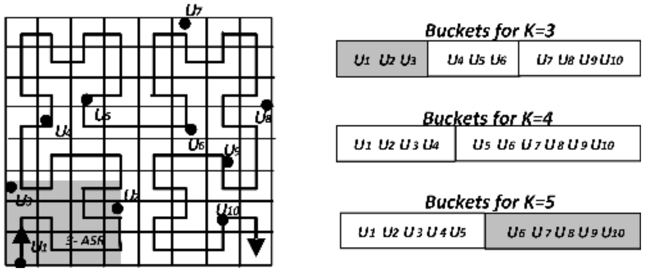


Figure 7: An example of *Hilbert Cloak*.

requirement of $\mathcal{K} = 2$, the returned ASR is the same cell. In the event that a query with the same anonymity requirements is issued from cell $\langle(1, 2), (2, 3)\rangle$ the returned ASR is the union of cells $\langle(1, 2), (2, 3)\rangle \cup \langle(1, 3), (2, 4)\rangle$.

Interval Cloak [8] is very similar to *Casper* as it also partitions the total area that is covered by the trusted server into equi–sized quadrants and organizes this information in a Quad–tree structure. However, *Interval Cloak* does not consider the neighboring cells at the same level when computing the ASR, but instead it directly ascends to the ancestor level in the pyramid. As an example, in Figure 6, a request for an LBS that is issued by U_3 or U_4 will generate the ASR $\langle(0, 2), (2, 4)\rangle$ (instead of $\langle(1, 2), (2, 4)\rangle$ for *Casper*). As is evident, *Casper* is more effective in producing compact ASRs when compared to *Interval Cloak*. However, as is proven in [9], both *Interval Cloak* and *Casper* are secure only for uniform data distributions.

Hilbert Cloak [9] does not suffer from this shortcoming as it generates the same $\mathcal{K}$–ASR, no matter who among the participants of the anonymity set requested the service. The proposed approach is based on $\mathcal{K}$–bucket cloaking; it dynamically arranges the users into groups of $\mathcal{K}$ and computes the ASR as the MBR enclosure that contains the $\mathcal{K}$ users in the group of the requester. *Hilbert Cloak* creates an one–dimensional mapping of the position of each user. In the proposed mapping, locations that are near each other in the two–dimensional plane, are expected to also lie near each other to its one–dimensional transformation. For each request with an anonymity requirement of $\mathcal{K}$, *Hilbert Cloak* partitions each $\mathcal{K}$ users in the system into a bucket according to their Hilbert values. Following that, *Hilbert Cloak* retrieves all the $\mathcal{K}$–1 users that lie in the same bucket as the requester, and formulates the $\mathcal{K}$–ASR as their MBR enclosure. An example of this operation is presented in Figure 7, where we consider 10 users whose IDs are sorted in ascending order based on their Hilbert values. Given a query for an LBS from U_3 with an anonymity requirement of $\mathcal{K} = 3$, *Hilbert Cloak* uses the rank of the user (here 3) to dynamically identify the bucket in which he/she is partitioned (here is the first bucket). Then, it retrieves all the users who are partitioned in the same bucket as the requester (i.e. U_1, U_2 and U_3) and returns their MBR as the computed ASR for this request (see the 3–ASR shaded region in Figure 7). Notice that any query with an anonymity requirement of $\mathcal{K} = 3$ originating from any of U_1, U_2, would generate the exact same 3–ASR as the one that is generated for U_3. Furthermore, it must be noted that *Hilbert Cloak* can generate the bucket that contains the requester on–the–fly based on the rank of each user in the system.

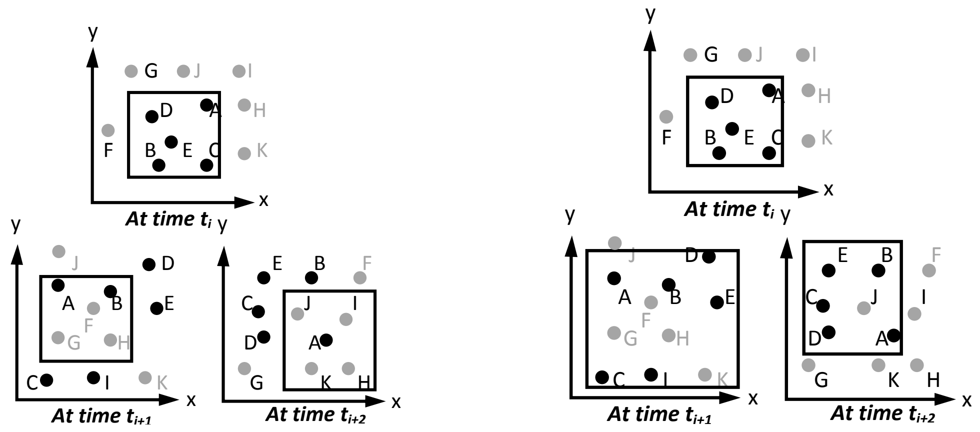
4.4 Historical K–anonymity

Historical approaches to $\mathcal{K}$–anonymity (e.g., [1,14,15]), keep track of the movement history of each user in the system and utilize this information when building the anonymity regions for the user requests. Compared to other methodologies for the offering of $\mathcal{K}$–anonymity in LBSs, in historical $\mathcal{K}$–anonymity approaches the participants of the anonymity set are selected based on their history of movement in the system, with the requirement that at some time in their history of movement these users were close to the point of request. Ref. [14] states this observation as "using users' footprints instead of their current positions, for cloaking".

Refs. [1, 15] consider the area that is covered by the trusted server as a set of Places–of–Interest (POIs) defined for each user in the system. Each POI has a spatial extent and can be related to an unanchored temporal interval. It represents a place that is frequently visited by a user based on his/her history of movement in the system, as well as the approximate time–of–day (represented as a time interval) of these visits. A series of POIs (along with their corresponding time intervals) that are frequently visited in sequel by a user can be considered as hazardous with respect to the privacy of the user when requesting LBSs, as such requests can easily disclose his/her identity.

The trusted server monitors the users to identify when they request LBSs from any of their POIs. When a user transmits a request for an LBS from one of his/her POIs, the trusted server computes an area along with a time interval that contains the requester, as well as $\mathcal{K}$–1 other users who happened at sometime in the past to pass by the location of the request. The computed area is said to be (historically) $\mathcal{K}$–anonymous as it protects the requester by guaranteing that his/her location cannot be identified with a probability that is larger than $1/\mathcal{K}$, among the other $\mathcal{K}$–1 users. The whole anonymization process is guided by a set of spatial and temporal constraints; the spatial constraints require that the generated region of anonymity is within some reasonable spatial bounds so as to allow the provision of the requested LBS, while adequately hindering the actual location of request. On the other hand, the temporal constraints impose a barrier on how back in time (starting from the point of request) should the history of movement of all the users in the system be searched, so as to retrieve the participants of the anonymity set. Following the computation of the anonymity region, the trusted server forwards the request containing the cloaked user location to the service provider for servicing. When a subsequent request is received from the same user, the trusted server tries to match this request to the next POI in his/her sequence of POIs, and if the match is successful it recomputes the area along with the time interval that contains the requester, as well as his/her $\mathcal{K}$–1 original neighbors.

Ref. [14] uses the history of movement of all the users in the system to apply a recursive top–down partitioning of the area that is covered by the trusted server into equi–size quadrants. Each cell of the partitioning is divided into quadrants up to the point that it contains at most N users who have visited this cell sometime in their movement history. By using this structure alongside a hash table that records the user IDs and the trajectories for the users of each cell, the proposed approach requires that each user who wishes to request an LBS has to first communicate a *base* trajectory to the trusted server. A base trajectory $T = \{c_1, c_2, \ldots, c_n\}$



(a) The attack scenario.

(b) Blocking the attack.

Figure 9: The query tracking attack and its elimination.

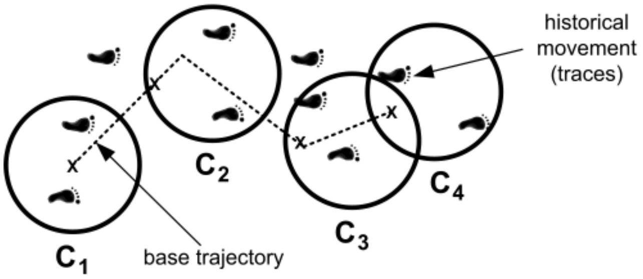


Figure 8: $\mathcal{K}$–anonymity based on historical movement.

defines the itinerary that the user will follow in the system, where each c_i corresponds to a location update on the trajectory on which the user will move. In response, the trusted server computes a new trajectory $T' = \{C_1, C_2, \ldots, C_n\}$ that provides $\mathcal{K}$–anonymity to the user when using the LBS throughout his/her declared itinerary. Each C_i in T' corresponds to a region of (historical) $\mathcal{K}$–anonymity that contains the requester (base trajectory) along with $\mathcal{K}$–1 users based on their historical trajectories in the system. Whenever the user arrives at c_i he/she informs the trusted server who, in turn, uses the $\mathcal{K}$–anonymity region of C_i to continue to protect his/her privacy. Figure 8 demonstrates the operation of the algorithm for the offering of historical 3–anonymity to a requester of an LBS based on his/her computed base trajectory.

4.5 Trajectory K–anonymity

Trajectory $\mathcal{K}$–anonymity approaches (e.g., [2, 6, 7]) are appropriate for preserving the privacy of the users who request LBSs that cannot be offered in just a single communication of the user with the service provider. As an example, consider a car navigation service, in which the current position of the user has to be communicated to the service provider for as long as the user travels to his/her destination (so that he/she receives updated directions). The approaches of this category are responsible for protecting the whole trajectory of the requester from the time of request until the service provision. Such services are called *continuous* and the corresponding requests are termed as *continuous queries.* It is important to mention that contrary to historical $\mathcal{K}$–anonymity approaches (such those of [1, 14]), which can also protect the user trajectory by providing $\mathcal{K}$–anonymity to the requester of LBSs, trajectory $\mathcal{K}$–anonymity approaches generate the $\mathcal{K}$–ASRs by utilizing the current instead of the historical movement of the users in the system to adequately cover up the trajectory of the requester. In what follows, (a) we motivate the necessity for trajectory $\mathcal{K}$–anonymity methodologies by discussing the limitations of location $\mathcal{K}$–anonymity approaches, and (b) we provide a partitioning of the trajectory $\mathcal{K}$–anonymity algorithms along two principal directions, as well as discuss some of the most prevalent approaches in each direction.

4.5.1 Why trajectory $\mathcal{K}$–anonymity? — Query tracking in LBSs

Location $\mathcal{K}$–anonymity approaches suffer from correlation attacks which prevent them from protecting the requesters of continuous queries. As indicated in [2], the identity of the requester can be easily revealed based on the participants of his/her anonymity set. Figure 9 demonstrates how this is possible. Imagine a query submitted at time t_i by user A for a continuous service. The applied location cloaking strategy (e.g., [8, 10]) generates the 5–ASR shown in Figure 9(a) that includes the requester along with four of his/her neighbors: B, C, D and E. As the user moves, he/she needs to transmit a new location update to the service provider for the continuation of the service provision. Thus, at time t_{i+1} the user sends a new query to the trusted server containing his/her new location. However, the users in the neighborhood of the requester have also moved in the meanwhile and thus the new 5–ASR that is produced by the employed location $\mathcal{K}$–anonymity algorithm (i.e. $\{A, B, F, G, H\}$) has only user B in common with the previously generated ASR. As a result, if an attacker knows the two ASRs he/she can safely conclude that the requester is either A or B, which significantly reduces the actual degree of anonymity that is offered to A from 1/5 to 1/2. In the next location transmission, at t_{i+2}, the identity of the requester is revealed since no other participant in his/her current ASR was also part of all the previous ASRs.

To alleviate from correlation attacks, existing approaches to trajectory $\mathcal{K}$–anonymity ensure that all the $\mathcal{K}$ participants of the first $\mathcal{K}$–ASR will also participate in all the subsequently computed $\mathcal{K}$–ASRs, as produced by the cloaking algorithm. This way of eliminating the query tracking attack is presented in Figure 9(b).

4.5.2 Generic approaches to trajectory K–anonymity

Generic approaches to trajectory $\mathcal{K}$–anonymity do not take into consideration any particular movement behavior of the requester of an LBS (as well as of the other users in the system), when providing him/her with trajectory $\mathcal{K}$–anonymity. Instead, all requests for LBSs are handled in exactly the same manner by the trusted server, no matter what the location of request is or the path that the user follows in the system during the LBS provision.

Ref. [2] proposes the first algorithm for trajectory $\mathcal{K}$–anonymity in LBSs. The main idea is to require that a user belongs in a group of at least $\mathcal{K}$–1 other users prior to sending a continuous query for an LBS. The generated $\mathcal{K}$–ASR in each location transmission of the user is computed as the MBR enclosure of all the users in the group of the requester, based on their location in the system. It is important to mention that while a request for an LBS is in progress, no grouped user that participated to the original anonymity set of the requester is allowed to leave the group, as this action would jeopardize the privacy of the requester.

4.5.3 Personalized approaches to trajectory K–anonymity

Personalized approaches to trajectory $\mathcal{K}$–anonymity utilize the history of movement of all the users in the system to cope with correlation attacks in continuous user queries. They differ from generic approaches to privacy in LBSs, primarily due to the following reasons: (a) they drop assumption 4 (Section 3) by consider attackers who have knowledge of the users' movement behavior in the system and can use their knowledge of the frequent movement patterns of the users to breach user privacy, (b) they depict the movement of each user u in the system as a continuous function $f(u, x, y, t)$, instead of a set of individual locations and times, (c) they automatically derive a set of frequent movement patterns per user based on his/her history of movement in the system, which are subsequently used to protect his/her privacy when requesting continuous LBSs, and (d) they can offer trajectory $\mathcal{K}$–anonymity to the requesters of LBSs by assuming an underlying network topology of user movement, instead of a grid–based, free–terrain solution.

Ref. [6] provides a trajectory $\mathcal{K}$–anonymity solution that uses the history of movement of the users in the system to derive a set of frequent mobility patterns per user. Each of these patterns corresponds to a route (instead of a sequence of POIs and related time periods, as in [1]) that is frequently followed by the corresponding user in the system and is stored as an $f(x, y, t)$ function, having both a spatial and a temporal extent. The proposed algorithm identifies those frequent routes of a user that are rarely followed by many other users in the system. These routes are termed as *unsafe* for this user, as they can disclose his/her identity when requesting LBSs from within any of them. In [6] a grid–based, free–terrain solution is employed that utilizes the computed unsafe routes of the users, in order to provide them with $\mathcal{K}$–anonymity when requesting LBSs.

In [7] a network–based privacy model is proposed that considers an underlying network of user movement in order to derive the unsafe routes of the users and to offer trajectory $\mathcal{K}$–anonymity to the requesters of LBSs. With respect to the offering of $\mathcal{K}$–anonymity, the proposed approach considers two spatial cloaking strategies, depending on the location of the requester at the time of request, as well as his/her subsequent locations until the provision of the service. In particular, $\mathcal{K}$*–present* (the so–called weak) trajectory anonymity identifies $\mathcal{K}$–1 users that are close to the requester at the time of request and thus could have issued the request for the LBS. On the other hand, $\mathcal{K}$*–frequent* (strong) trajectory anonymity, collects the subjects who were near the requester at the time of request and for whom the currently traveled route of the requester is also frequent.

5. CONCLUSIONS

In this paper, we presented a survey on the state–of–the–art centralized $\mathcal{K}$–anonymity approaches for the offering of privacy in LBSs. The aim of the presented methodologies is to protect the location of the requesters of LBSs in both static and continuous queries. On the other hand, a new and very prominent body of research regards $\mathcal{K}$–anonymity methodologies that protect the content of the user query in addition to the location of the user. We believe that future work in this research direction will lead to more robust and thorough methodologies that better protect the privacy of the user when requesting LBSs.

6. REFERENCES

[1] C. Bettini, X. S. Wang, and S. Jajodia. Protecting privacy against location–based personal identification. In *Proceedings of the 2nd VLDB Workshop on Secure Data Management*, pages 185–199, 2005.

[2] C. Y. Chow and M. F. Mokbel. Enabling private continuous queries for revealed user locations. In *Proceedings of the 10th International Symposium on Advances in Spatial and Temporal Databases (SSTD)*, pages 258–275, 2007.

[3] R. Clarke. Person location and person tracking — technologies, risks and policy implications. *Information Technology and People*, 14(2):206–231, 2001.

[4] J. Domingo-Ferrer, editor. *Inference Control in Statistical Databases: From Theory to Practice*, volume 2316 of *Lecture Notes in Computer Science*. Springer, 2002.

[5] B. Gedik and L. Liu. Location privacy in mobile systems: A personalized anonymization model. In *Proceedings of the 25th IEEE International Conference on Distributed Computing Systems (ICDCS)*, pages 620–629, 2005.

[6] A. Gkoulalas-Divanis and V. S. Verykios. A free terrain model for trajectory $\mathcal{K}$–anonymity. In *Proceedings of the 19th International Conference on Database and Expert Systems Applications (DEXA)*, pages 49–56, 2008.

[7] A. Gkoulalas-Divanis, V. S. Verykios, and M. F. Mokbel. Identifying unsafe routes for network–based trajectory privacy. In *Proceedings of the SIAM International Conference on Data Mining (SDM)*, 2009.

[8] M. Gruteser and D. Grunwald. Anonymous usage of location–based services through spatial and temporal cloaking. In *Proceedings of the 1st International Conference on Mobile Systems, Applications, and Services (MOBISYS)*, pages 31–42, 2003.

[9] P. Kalnis, G. Ghinita, K. Mouratidis, and D. Papadias. Preventing location–based identity inference in anonymous spatial queries. *IEEE Transactions on Knowledge and Data Engineering*, 19(12):1719–1733, 2007.

[10] M. F. Mokbel, C. Y. Chow, and W. G. Aref. The new Casper: query processing for location services without compromising privacy. In *Proceedings of the 32nd International Conference on Very Large Data Bases (VLDB)*, pages 763–774, 2006.

[11] P. Samarati. Protecting respondents' identities in microdata release. *IEEE Transactions on Knowledge and Data Engineering*, 13(6):1010–1027, 2001.

[12] P. Samarati and L. Sweeney. Protecting privacy when disclosing information: $\mathcal{K}$–anonymity and its enforcement through generalization and suppresion. In *Proceedings of the IEEE Symposium on Research in Security and Privacy (SRSP)*, pages 384–393, 1998.

[13] H. Samet. *The Design and Analysis of Spatial Data Structures*. Addison–Wesley Longman Publishing Co., Inc., 1990.

[14] T. Xu and Y. Cai. Exploring historical location data for anonymity preservation in location–based services. In *Proceedings of the 27th IEEE Conference on Computer Communications (INFOCOM)*, pages 547–555, 2008.

[15] P. Zacharouli, A. Gkoulalas-Divanis, and V. S. Verykios. A $\mathcal{K}$–anonymity model for spatiotemporal data. In *Proceedings of the IEEE Workshop on Spatio–Temporal Data Mining (STDM)*, pages 555–564, 2007.