Privacy–Preserving Trajectory Collection

Győző Gidófalvi*
Uppsala University
Dept. of Information Tech.
gyozo.gidofalvi@it.uu.se

Xuegang Huang
Aalborg University
Dept. of Computer Science
xghuang@cs.aau.dk

Torben Bach Pedersen
Aalborg University
Dept. of Computer Science
tbp@cs.aau.dk

ABSTRACT

In order to provide context–aware Location–Based Services, real location data of mobile users must be collected and analyzed by spatio–temporal data mining methods. However, the data mining methods need precise location data, while the mobile users want to protect their location privacy. To remedy this situation, this paper first formally defines novel location privacy requirements. Then, it briefly presents a system for privacy–preserving trajectory collection that meets these requirements. The system is composed of an untrusted server and clients communicating in a P2P network. Location data is anonymized in the system using data cloaking and data swapping techniques. Finally, the paper empirically demonstrates that the proposed system is effective and feasible.

Categories and Subject Descriptors

H.2.8 [**Information Systems**]: Database Applications—*Data mining, Spatial databases and GIS*

General Terms

Algorithms

Keywords

Privacy, anonymity, diversity, data swapping, data cloaking, data mining, moving object trajectories, LBS, P2P

1. INTRODUCTION

For a Location–Based Service (LBS) to be effective, it must be context–aware. To support this, location data of mobile users must be collected and data mined for spatio–temporal patterns. However, the data mining methods need precise location data, while the mobile users want to protect their location. Consider the following application scenario: A large shopping mall wants to do data mining on customer activities, including where and when customers were in the mall, and their movement patterns while there. The collection assumes a positioning system, e.g., GPS, radio–locationing, or RFID, for detecting the locations of the customers' mobile phones. The phones record locations and send them to a central server for analysis. The analysis can, e.g., reveal cross–selling opportunities that can form the basis of personalized Location–Based Advertising (LBA). However, to preserve the location privacy of users, the trajectories must be collected in an anonymized way.

Locations of an individual can serve as a quasi–identifier when linked with other public spatial data sources [1, 6]. However, this is *not* an issue in the proposed solution, as the intended use is in a public place free of "sensitive locations." Nevertheless, trajectories can contain "embarrassing" locations, e.g., an adult video store, or "secret" locations, e.g., a jewelery shop, which the user does not want to be associated with/linked to. Thus, a collection system that collects *exact* user trajectories, but hides the identities of the users so that they *cannot be linked to their trajectories* is highly needed.

The herein proposed collection system has been specifically designed to work with clients communicating through *P2P network protocols* such as Bluetooth or ZigBee. In comparison to "routed" protocols (GPRS, 3G, or Wi–Fi), the advantages of P2P protocols is free communication and low power consumption. On the other hand, P2P protocols expose a hardware ID (like a MAC address), a disadvantage given the use–scenario as this exposure allows the recipient to directly link the identity of the sender to the data being transmitted, leading to a possible breach of privacy.

There exist a good deal of related previous work. Within data mining on trajectory data, methods have been proposed to extract location or movement patterns [5, 7]. Such methods can benefit a lot from the exact data collected by the proposed method. General methods for privacy–preserving data mining has become important, see [11] for an overview. The proposed solution uses specially tailored versions of two common strategies for privacy–preserving data mining, namely data *cloaking* and data *swapping*

With the rapid emergence of LBSes, location privacy has also become an important topic. As locations themselves can be quasi–identifiers [1, 6], several papers have proposed privacy–preserving LBS solutions [1, 2, 3, 4, 6, 8, 12]. These solutions aim to provide privacy either by spatio–temporal generalization or by mixing exact user locations with dummy locations. Spatio–temporal generalization can be based on a minimum area requirement and/or a k–anonymity requirement, which is an extension of the general k–anonymity model [10], whereby the locations of k users is aggregated so that the location of an individual user is not distinguishable [3, 4]. While these privacy–preserving LBS solution could be modified to collect trajectories in a privacy–preserving manner, such modifications would entail at least one of the following: 1)

*Contact author. This work was, in large part, performed during a period when the contact author's primary affiliation was with Geomatic ApS / Aalborg University, and was supported in part by the Danish Ministry of Science, Technology, and Innovation under grant number 61480.

ACM GIS '08, November 5-7, 2008. Irvine, CA, USA

system components (server or clients) must be trusted, 2) the collected trajectories are blurred or noisy, on which data mining does not yield accurate results [6]. In comparison, the herein proposed trajectory collection system: 1) does *not have trusted components*, and 2) is able to *collect exact trajectories without loss and noise*.

The contributions of this paper are as follows. First, the paper adapts general privacy definitions to derive location privacy definitions of varying strengths. Second, the paper proposes a complete system for the *lossless* collection of *exact* trajectories in a privacy–preserving manner according to the definition. Finally, the paper empirically evaluates and shows the effectiveness and feasibility of the proposal under reasonable conditions and privacy settings.

2. PROBLEM DEFINITION

Denote each moving object as a *data item*, $di = (id, S)$, where id is the identity attribute value, or ID of the data item and S is the trajectory of di. S is modeled as a timestamped sequence of locations. A *trajectory piece* is defined as a subset of a trajectory ordered on the time domain. For instance, $S_1 = \langle(loc_1, t_1), \ldots, (loc_k, t_k)\rangle$ and $S_2 = \langle(loc_{k+1}, t_{k+1}), \ldots, (loc_n, t_n)\rangle$ are two trajectory pieces of $S = \langle(loc_1, t_1), \ldots, (loc_n, t_n)\rangle$. For a given data item $di = (id, S)$, both $di_1 = (id, S_1)$ and $di_2 = (id, S_2)$ (with $di_1.id = di_2.id = di.id$) represent part of the trajectory of di. di_1 and di_2 are called *partial data items*, or `pdis` for short.

Problem Statement: In the proposed system, sets of `pdis` are exchanged within the P2P network. During an exchange, the fixed hardware ID, hid_A, of the sender A is revealed to the recipient B. It is a breach of privacy if B can, with a higher than desired probability, infer that the trajectory piece S_j in a received `pdi`, $di_j = (id, S_j)$, describes the actual partial movement of A. In relation to traditional privacy protection concepts, the identity attribute id is the quasi–identifier, and the trajectory piece S_j is the sensitive attribute. The aim of the proposed system is to prevent B from linking hid_A to the true identifier id^A used by A (thereby linking hid_A to S_j) with a probability above a certain threshold.

To establish a framework for avoiding the above privacy breach, the following definition adapts the traditional notion of *k*–anonymity [10] to the problem setting.

DEFINITION 1. *(*k–anonymity of data items*) For a set of moving objects* $\mathcal{MO} = \{o_1, \ldots, o_m\}$ *and a set of data items* $\mathcal{DI} = \{(id_1, S_1), \ldots, (id_n, S_n)\}, m \leq n$ *where* $id_1, \ldots, id_n$ *are encoded identity values and* $S_1, \ldots, S_n$ *are trajectory pieces, the moving objects and the data items are said to preserve* k–anonymity *if both of the following conditions are true:*

1. *For any object o, there are at least k data items* $(id'_1, S'_1), \ldots, (id'_k, S'_k)$ *that correspond to o with equal probability.*
2. *For any data item* (id, S)*, there are at least k objects* $o'_1, \ldots, o'_k$ *that correspond to* (id, S) *with equal probability.*

As it is pointed out in [6], when trajectories of k moving objects are almost identical, *k*–anonymity does not adequately protect the privacy of users. It is necessary to require that the k trajectories possess a certain spatial diversity. The following definition of α*–diversity* adopts the notion of l–diversity [9] to the spatial and spatio-temporal domain, which is inherent in the problem setting.

DEFINITION 2. *(*α–diversity of data items*) For a set of data items* $\mathcal{DI} = \{(id_1, S_1), \ldots, (id_n, S_n)\}$ *where* $id_1 \neq id_2 \neq \ldots \neq id_n$*, and a given threshold* α*, these data items preserve* α–diversity *if* $AREA(MBR(\{S_1, \ldots, S_n\})) \geq \alpha$*, where* MBR *is the minimum bounding rectangle of* $S_1, \ldots, S_n$ *and* AREA *returns the area size of an MBR.*

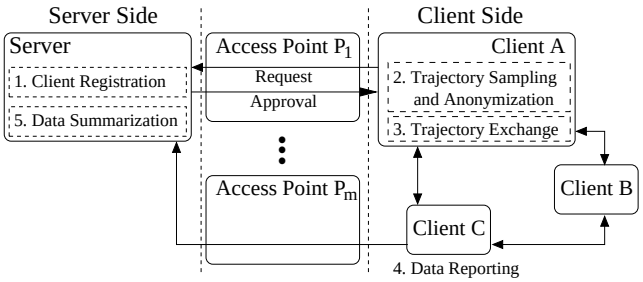


Figure 1: System Architecture.

Next, combining the definitions of *k–anonymity* and α*–diversity*, a more strict setting to protect the privacy the data items is presented, namely the *k–*α*–anonymity*.

DEFINITION 3. *(*k–α–anonymity of data items*) For a set of moving objects* $\mathcal{MO} = \{o_1, \ldots, o_m\}$ *and a set of data items* $\mathcal{DI} = \{(id_1, S_1), \ldots, (id_n, S_n)\}, k \leq m \leq n$ *that preserve* k–anonymity*, and a threshold value* α*, the moving objects and data items satisfy* k–α–anonymity *if the following condition is true. For any k data items* $(id'_1, S'_1), \ldots, (id'_k, S'_k) \in \mathcal{DI}$*, these data items preserve* α–diversity.

The aim of the herein proposed data collection solution is to preserve anonymity of data items (or partial data items) in any set of data being stored, transmitted or collected in the system.

3. SOLUTION

The following section describes the system architecture, the five stages of the anonymization process, and discusses how and with what strength privacy guarantees are achieved during the stages.

System Architecture and Anonymization in Five Stages: The proposed system involves a server side application installed at a central server and a client side application that is installed at each mobile phone. The client side application is able to communicate with other client applications through a P2P network such as Bluetooth. The client side application can also communicate indirectly with the server via P2P access points. Consequently, the server can communicate with clients by broadcasting through these P2P access points for an appropriate period. Each client side application is called a *client* and the server side application is called the *server*. The server includes two processes, *server_reg* and *server_sum*, which are kept running during the whole lifetime of the server. The client, once activated, starts a single process, *client_proc*. The top–level algorithm of *client_proc* is listed below.

(1) **procedure** *client_proc* $(server, hid, k, \alpha, \lambda, L)$
(2) $(T_s, \tau, \tau_{max}) \leftarrow$ *register*$(server, hid, k)$
(3) $id_1, id_2, \ldots, id_k \leftarrow$*gen_hashID*$(hid, now(), k)$
(4) **while** $now() \in [T_s, T_s + \tau)$
(5) $DB \leftarrow$*sample_anonymize*$(\{id_1, \ldots, id_k\}, k, \alpha, \lambda)$
(6) *exchange*$(k, DB,$ *type* $= MUTUAL)$
(7) $t_{old} \leftarrow oldest(DB)$
(8) **if** $(size\_of(DB) \geq L) \cup (now() - t_{old} \geq \tau_{max})$
(9) *report*$(server, DB)$
(10) **if** $size\_of(DB) \leq k \times num\_of\_peers(k)$
(11) *exchange*$(k, DB,$ *type* $= SEND)$
(12) **else** : *report*$(server, DB)$
(13) **return** Q_{dp}

Among the parameters of *client_proc*, $server$ and hid specify the server address and the hardware ID of the client, and k, α, λ, L are kept in the client's profile. The values k, α specify the user's requirement on *k–*α*–anonymity*. λ and L are described in the follow-

ing. In the pseudo code of *client_proc*, the variable *DB* is a database of all trajectory pieces in the current instance of *client_proc*.

With the three processes, *server_reg*, *server_sum* and *client_proc*, the whole system can be summarized into five stages, namely: 1) *client registration*, 2) *trajectory sampling and anonymization*, 3) *trajectory exchange*, 4) *data reporting*, and 5) *data summarization*. As depicted in Figure 1, at stage 1, client A sends a registration request to the server (line 2 of *client_proc*). The server process (*server_reg*) accepts the request of A and the requests of $k-1$ other clients and assigns the time and period that the k clients should start and run the other steps of *client_proc*. Upon receiving the approval from the server, client A starts stage 2. At this stage, client A generates a set of k different ID values (line 3 of *client_proc*). One of these ID values is selected as the ID of the actual trajectory and the others are used as the ID values of cloaking trajectories (explained later). The cloaking trajectories with these ID values are called the *cloaking (trajectory) data*. Client A begins to sample its trajectory and generate the cloaking trajectory data (line 5 of *client_proc*). The parameter λ of *client_proc* specifies the time interval for cutting the sampled trajectory and the cloaking data into trajectory pieces. Then, at stage 3, as shown in Figure 1, through the P2P network, client A communicates with clients B and C to exchange the anonymized data (line 6 of *client_proc*). The *trajectory sampling and anonymization* and *trajectory exchange* steps are kept running until the period specified by the server is reached (the while–loop of *client_proc*). At stage 4, when a client has reached its storage limit (specified by the parameter L of the *client_proc*) or if it has a very "old" piece of trajectory data, this client reports its data to the server (line 9 of *client_proc*). The stage of *data reporting* also happens when clients are at the end of the activation period (lines 10–12 of *client_proc*). In Figure 1, clients A and B choose to transmit their collected data to other clients in the P2P network (line 11 of *client_proc*) and client C sends its collected data to the server (line 12 of *client_proc*). Finally, at stage 5, the server process *server_sum* summarizes the data from the clients, filters out the cloaking trajectory data and computes the real trajectory data. The next subsections further describe details of these stages and describe their privacy features.

Client registration: In this stage, the client sends a registration request to the server. As a response, the client receives timing parameters from the server about when to invoke subsequent stages, in particular the *data reporting* stage. To guarantee *k–anonymity* of clients at the server side, the server maintains a FIFO buffer of incoming requests, and dispatches k clients with the same timing parameters. As a result, even if $k-1$ of the clients, dispatched simultaneously, "abort" without invoking the *data reporting* stage, the *k–anonymity* of the remaining client is preserved.

Trajectory Sampling and Anonymization: In this stage, a set of k different ID values, $id_1, \ldots, id_k$, are generated at the client. To ensure that the k values are unique among the other ID values generated at different instances of the client, in the *gen_hashID* function (line 3 of *client_proc*), the Secure Hash Algorithm is used. Then, the client associates one of the ID values, but without loss of generality the first ID value, id_1, with its real trajectory.

Then, in a continuous fashion the client starts to sample its real trajectory, S_1, and generate subsequent locations of $k-1$ cloaking trajectories, $S_2, \ldots, S_k$, at every unit time instance. Then, at every time interval λ, the client cuts the trajectories and gets trajectories pieces, $S_1^{'}, \ldots, S_k^{'}$, for the time interval $[now() - \lambda, now())$. $(id_1, S_1^{'})$ is referred to as a *sampled*–pdi, or s-pdi, while the other $k-1$ trajectory pieces together with their corresponding IDs are referred to as *generated*–pdis, or g-pdis for short.

To distinguish the s-pdi among all the pdis, the client generates and stores in *DB* an *even* number of copies of the s-pdi and an *odd* numbers of copies of a g-pdi. The idea of having copy amounts of the pdis with different parity is to hide the s-pdis in exchanges between clients, but to be able to identify them at the server. Specifically, given an instance of k pdis, one with an even and $k-1$ with an odd number of copies, it is impossible for a client holding some, but not necessarily all, copies of a pdi to decide whether the pdi is sampled or generated.

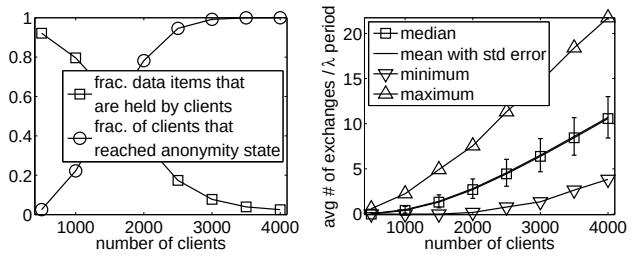
Subsequent locations of cloaking trajectories, $S_2, \ldots, S_k$, are generated in two steps. Step 1 is executed once at the start of every λ–period, and ensures that sampled and the generated locations are pairwise α–diverse. Step 2 is executed multiple times during a λ–period, and starting from the pairwise α–diverse locations generated in Step 1, subsequent locations in the trajectories are generated essentially randomly, but obeying some spatial constraints, which, among others, mimic the speed of the of the user.

When the pdis are added to the trajectory database *DB*, the MBR of all the trajectory pieces has an area size bigger than α. Since, at this time, there are at least k pdis in the *DB* and these pdis have k different ID values, all the pdis in *DB* preserve *k–anonymity* as well as *α–diversity*.

Trajectory Exchange: In this stage, the client starts to exchange the pdis in its *DB* with other clients in the P2P network. To satisfy the first condition in Definition 1, considering possibly multiple exchanges between two clients, during an exchange it must be true that any pdi selected for exchange must come from a pool of pdis in which, at least potentially, for at least $k-1$ IDs the number of pdis are statistically equal to the number of pdis for id_1, the ID of the real trajectory of the client. To guarantee the above, the client selects s- and g-pdis with equal probability for exchange from the pool of pdis in *DB*, which are accumulated during the *trajectory sampling and anonymization* step. Notably, if no cloaking trajectories would be generated, *DB* would only contain pdis for id_1, preventing any exchange to take place. Hence, the generated cloaking trajectories are also necessary to facilitate a bootstrap mechanism for the system. To satisfy the second condition in Definition 1, a client A can only send data to a client B, if B has at least $k-1$ other clients in its neighborhood. The process of obtaining the hardware IDs A's neighbors that satisfy the above condition is referred to as Neighborhood Detection (ND). Since clients cannot trust each other, A cannot perform ND by asking its neighbors for the number of their respective neighbors. Instead, A asks from each of its neighbors for the hardware IDs of their respective neighbors. Then, by combining and aggregating the answer sets, client A probabilistically verifies each of the answers.

To satisfy Definition 2, the client ensures that at least two s- or g-pdis with different IDs are selected for exchange. Finally, to further hinder a malicious clients B from even linking the hardware ID of the sender A to the *set* of IDs used by A, A selects additional pdis for exchange from the set of s- and g-pdis *and* the set of pdis received in previous exchanges, referred to as *exchanged*–pdis, or e-pdis for short. All aspects of the selection process are effectively facilitated by storing s/g-pdis and e-pdis in two priority queues, respectively, favoring the oldest pdis with the most number of copies for selection. As a result of the careful design of the above two stages, pdis in an exchange preserve *k–α–anonymity*.

Data Reporting: In this stage, the client reports a subset S of the pdis in its *DB* to the server. To satisfy the first condition in Definition 1, the client constructs S such that the number of pdis for each ID in S is statistically equal. Furthermore, to minimize the



(a) Anonymity. (b) Number of Exchanges.

Figure 2: Evaluation for $k = 5$, $r = 10$m and Varying p.

amount and to control the age of the data that is potentially not reported, the client constructs S such that 1) S is *maximal* w.r.t. the IDs present in *DB*, and for each ID, in case of multiple, the oldest pdi is included in S. In practice, S consists of the oldest pdi for every ID that is present in *DB*. S is referred to as the *maximal anonymity set*. When $|S| \approx |DB|$, the *DB* and the respective client are said to be in *anonymity state*. To satisfy the second condition in Definition 1, the client sends S to the server only under the same condition as when a exchange would be possible. As, through the previous stages, the client can guarantee the α*–diversity* of *DB*, the maximal anonymity set, S, which is reported to the server preserves k*–*α*–anonymity*.

Data Summarization: In this stage, as a continuous process, the server summarizes the reports as follows. The server stores and arranges the received pdis into groups at each λ–period based on the ID values. To be able to determine when all pdis should have been received for a particular ID, id_i, the server records the time, t'_i, when the first pdi was received for id_i. At (or after) time $t'_i + 2\tau_{max}$ the server checks the parities of the pdis at each λ–period for id_i. If the majority of the parities are even, the server merges the trajectory pieces based on the timestamps of the trajectory pieces and stores the whole trajectory S_i. If the majority of the parities are odd, the group is removed as it represents cloaking trajectory data. The majority check is performed as a step to correct for the very few trajectory pieces that are lost, i.e., are not reported.

4. EMPIRICAL EVALUATION

In the absence of large, spatio–temporally dense, real–world data sets that record the pairwise wireless connectivity between mobile devices, to empirically evaluate the effectiveness of the proposed method, a grid–based shopping movement simulator was developed to generate realistic movements of clients in a $177,200\text{m}^2$ large shopping mall with 16 shops. In the simulation, clients move from a 5×5–meter grid cell to a neighboring cell at a speed of 1 m/s. A simulation step is 5 seconds long, i.e., a client can only move at most one grid cell in a simulation step. The movements of clients are random, but clients 1) obey obstacles, 2) try to follow a randomly selected direction if possible, 3) avoid re–visiting shops, and 3) slow down inside shops mimicking browsing behavior.

The effectiveness and feasibility of the proposed method was evaluated using four groups of measures: anonymity, age of the oldest data item, number of exchanges, and P2P communication characteristics. Three groups of 20–minute long simulations were performed varying the number of clients p, the anonymity parameter k, and the communication range r in the P2P network. The λ–period in the simulations was 60 seconds, during which each client collected $n \in \{2, 3\}$ copies of k s/g-pdis, and performed 4 NDs in order to find other clients to exchange data with.

Figure 2 shows two out of the four measures for $k = 5$, $r = 10$m and varying $p = [500, 4000]$. As p increases, the number of exchanges increases, see Figure 2(b), which allows more and more clients to reach anonymity state and leaves less and less fraction of the data on the clients, where it might get lost, see Figure 2(a). While it is not shown in Figure 2, the clever pdi–selection heuristic ensures the latter data contains only very recent s/g-pdis. This potential data loss is fully corrected for in the *data summarization* stage. W.r.t. P2P communication characteristics, although not shown in Figure 2, the experiments show that 1) the average neighborhood size is $[0.5, 4.3]$, 2) over 90% of the answers to NDs can be verified with high probability, and 3) for larger p values most of the NDs during a λ–period are effective, i.e., allow exchange(s). The experiments for varying $r = [5, 25]$ and $k = [2, 10]$ show similar results with similar and inverse trends, respectively. In summary, the experiments reveal that the proposal is effective in terms of anonymization and feasible in terms P2P communication.

5. CONCLUSIONS AND FUTURE WORK

The paper considered the problem of collecting trajectories of moving objects in a privacy–preserving manner. As a premiss for studying the problem, the paper first adapted and combined general data privacy definitions to derive definitions for location privacy. Then, to solve the problem, the paper proposed a *scalable* and *robust*, *complete* system, one which does *not* require trusted components, for the *lossless*, *privacy–preserving* collection of *exact* trajectories. Finally, the paper empirically demonstrated that the proposed system is effective and feasible under reasonable conditions and privacy/anonymity settings.

Future work will be along four paths. First, different kinds of system architectures will be considered. For example, the addition of hotspots to the P2P network will be considered. These hotspots will further reduce the chance of isolated clients and could potentially be used to distribute the *client registration* and *data summarization* stages. Second, future work will consider the implementation and the large–scale real–world deployment of a full system. Third, as an alternative to the verification-based, probabilistic ND method, more secure methods for performing anonymous ND in a P2P network will be considered. Finally, theoretical work will consider 1) proving the robustness of the system, and 2) finding a theoretical model of the system performance in relation to privacy.

6. REFERENCES

[1] C. Bettini, X. S. Wang, and S. Jajodia. Protecting Privacy Against Location–Based Personal Identification. In *Proc. of the VLDB Workshop on Secure Data Management, SDM*, 2005.
[2] C. -Y. Chow, M. F. Mokbel, and X. Liu. A Peer–to–Peer Spatial Cloaking Algorithm for Anonymous Location–based Services. In *Proc. of ACM GIS*, 2006.
[3] M. Gruteser and D. Grunwald. Anonymous Usage of Location–Based Services Through Spatial and Temporal Cloaking. In *Proc. of USENIX Mobisys*, 2003.
[4] B. Gedik and L. Liu. Location Privacy in Mobile Systems: A Personalized Anonymization Model. In *Proc. of ICDCS*, 2005.
[5] G. Gidófalvi and T. B. Pedersen. Mining Long, Sharable Patterns in Trajectories of Moving Objects. In *Proc. of STDBM*, 2006.
[6] G. Gidófalvi, X. Huang, and T. B. Pedersen Privacy–Preserving Data Mining on Moving Object Trajectories. In *Proc. of MDM*, 2007.
[7] C. S. Jensen, D. Lin, B. C. Ooi, and R. Zhang. Effective Density Queries on Continuously Moving Objects. In *Proc. of ICDE*, 2006.
[8] M. F. Mokbel, C. -Y. Chow, and W. G. Aref. The New Casper: Query Processing for Location Services without Compromising Privacy. In *Proc. of VLDB*, 2006.
[9] A. Machanavajjhala, J. Gehrke, et. al. L–Diversity: Privacy Beyond k–Anonymity. In *Proc. of ICDE*, 2006.
[10] L. Sweeney. K–Anonymity: A Model for Protecting Privacy. In *Int. J. of Uncertainty, Fuzziness and Knowledge–Based Systems*, 10(5), 2002.
[11] V. S. Verykios, E. Bertino, et. al. State–of–the–art in Privacy Preserving Data Mining. In *SIGMOD Record*, 33(1), 2004.
[12] T.-H. You, W.-C. Peng, and W.-C. Lee. Protecting Moving Trajectories with Dummies. In *Proc. of PALMS*, 2007.