PRESERVING PRIVACY FOR LOCATION-BASED SERVICES WITH CONTINUOUS QUERIES

YIMING WANG

A THESIS

IN

THE CONCORDIA INSTITUTE FOR INFORMATION SYSTEMS ENGINEERING

PRESENTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF MASTER OF APPLIED SCIENCE IN INFORMATION SYSTEMS SECURITY

CONCORDIA UNIVERSITY

MONTRÉAL, QUÉBEC, CANADA

APRIL 2010

Your file Votre référence
ISBN: 978-0-494-67274-7
Our file Notre référence
ISBN: 978-0-494-67274-7

ABSTRACT

Preserving Privacy for Location-Based Services with Continuous Queries

Yiming Wang

Location-based service (LBS) is gaining momentum as GPS-equipped mobile devices become increasingly affordable and popular. One of the potential obstacles faced by LBS is that users may raise concerns about their personal privacy when location data are sent to a distrusted LBS provider. A well-known solution is to render the location data less accurate through spatial or temporal cloaking. However, such a solution has limitations when the LBS is based on location data that either include speed and heading direction, or are sent at a regular time interval. In the former case, by combining consecutive location data including speed, heading direction, and cloaked locations, an adversary can obtain more accurate estimation of the actual location. In the latter case, an adversary can infer additional information when an expected update to the location data is not received because cloaking is not possible. In this thesis, we will first show how privacy protection provided by spatial cloaking can be breached, and proposed a new cloaking method to integrate the speed and direction into the spatial cloaking process. We then propose an auditing system to ensure all the mobile devices can be well protected even when it is impossible to cloak some of them to meet their customized privacy requirements. We evaluate the proposed methods

with experiments based on simulated mobile devices using real city maps. The experiments show that our speed and direction cloaking methods can achieve sufficient privacy protection without causing significant decrease in the service quality.

Acknowledgments

Firstly, I sincerely express my thanks to Dr.Lingyu Wang, who is my supervisor, constantly teach me and give me a lot of helps throughout my entitle research at Concordia University.

I thank also all of the professors I had during the course portion of this program from whom I learned immensely and drew inspiration to pursue this academic endeavor.

I would also like to thank Xue Kong for her contribution to the early stage of this research, and also like to appreciate all anonymous reviewers for their valuable comments on the thesis.

Finally, I would like to give my special thanks to my wife Jieli An, who encourages me all the time to finish the work, and also give my best love to my new born son, Yuchen Wang.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The Global Positioning System (GPS) is becoming a built-in feature of many wireless mobile devices, such as cellular phones and PDAs. GPS enables a device to send realtime location data including the speed, heading direction, and coordinates to a service provider. In exchange, the provider can offer customized services such as destination path, real time traffic jam, that are based on the device's location, namely, location-based services (LBS). LBS allows a mobile device to only receive and process data that are the most pertinent to the user's needs, and hence improves the device's performance and the user's experience.

One obstacle faced by LBS is the privacy concern that a distrusted LBS provider may learn users' personal habits or preferences from collected location data, and then abuse such information later on. A well-known solution is the spatial-temporal cloaking approach [21]. The accuracy of a device's location data is reduced (either spatially or temporally) such that the cloaked location is valid for at least k devices (where k is a predefined anonymity set size). The user may then feel his/her privacy is protected since an adversary can no longer determine which of the k users originates a query. Such an approach, however, cannot always succeed to protect every device while satisfying the device's constraint with respect to the accuracy of location-based services [13]. The issue can partially be addressed with temporal cloaking algorithms [13, 21, 47], although these are not an option for real-time

services that prevent the time delay required by temporal cloaking algorithms [30, 33].

In this thesis, we study the case of a moving device that sends in a sequence of LBS queries containing location data, such as coordinates, speed, and heading direction. For example, assume a car driver uses a mobile device and LBS to find an alternative route out of the traffic jam. Both the traffic status and the car's speed may vary significantly on different streets at different time. The LBS provider can anticipate the car's future location and thus provide more accurate results, if it has the car's location, speed, and heading direction at regular time intervals. We show that existing cloaking techniques including those especially designed for continuous queries (without speed and direction) become ineffective in such a case. An adversary may obtain more accurate estimations of the actual location than what is perceived from the cloaked locations. Such estimation may then allow him/her to violate the privacy constraint.

Moreover, we show that introducing more inaccuracy into the cloaked location cannot solve the problem. We also point out that most current cloaking methods cannot ensure all the devices requesting for LBS can be protected at their desired anonymity level while satisfying their constraints about the accuracy of LBS, especially when the mobile devices request real-time services or when the devices send their requests at a fixed time interval. We show that blocking a request that cannot be successfully cloaked from sending to the LBS provider is not effective in that the adversary still can breach the privacy protection based on the fact of a missing request.

We then propose a solution to prevent the aforementioned inference attack on location cloaking. The key idea is to simulate what an adversary can deduce, and then cloak not only the coordinates but also the heading direction and speed in order to prevent the inference. The cloaking depends on future locations, and the cloaking result must be disclosed before we know such locations. Therefore, the future locations must be approximately estimated based on the known parameters(speed, direction).

To address the issue of inferences caused by unsuccessful cloaking, we apply the simulatable auditing concept to location-based services to ensure all the mobile devices can be protected under their anonymity requirements. The key idea is to group the unsuccessfully cloaked devices together to achieve the anonymity protection. Since the number of such devices may be limited, the auditing process may also need to deny some additional successfully cloaked devices in order to protect the unsuccessful ones, which is an unavoidable tradeoff between the privacy and the quality of services [16]

The rest of the thesis is organized as follows. The rest of this section reviews necessary background knowledge about LBS, its privacy issues, location cloaking methods, and simulatable auditing. Section 2 reviews related work. Section 3 models the inference attack on existing coordinate-based location cloaking methods and define basic notions and notations. Section 4 introduces the proposed approach of cloaking both speed and heading direction in addition to coordinates of devices in order to prevent adversarial inferences. Section 5 describes the proposed auditing system in order to prevent adversarial inferences when some devices cannot be successfully cloaked. Section 6 presents simulation results based on randomly generated devices on real maps. Finally, Section 7 concludes the thesis and gives future directions.

1.1 Location-Based Services (LBS)

Wireless carriers and their partners are providing or developing new products, services, and business models that utilize location data. Location services provide information specific to a client device's location and they offer many opportunities to both users and service providers. For the mobile user, examples of location-based services include requesting the nearest businesses or services of a certain type, such as an ATM or Chinese restaurants, receiving alerts, such as notifications of a traffic jam, finding another user in the same region, etc. A big advantage is that mobile users do not need to manually specify their

Zip codes or other location identifiers in order to use location-based services. Also, the mobile devices do not need to be loaded with all the necessary information, such as maps or directories, which is important to devices with limited memory and storage. For the carrier, location-based services allow them to provide value-added services such as resource tracking with passive sensors or RFID tags on packages or train boxcars, finding people by skills (e.g., doctors), navigation through traffic jam, real-time weather, room schedules, proximity-based notification with targeted advertising or profile matching (e.g., dating), and payment based upon proximity (e.g., EZ pass).

The architecture of LBS usually involves three parties. A user or customer's device sends a query to a network operator or wireless service provider, which forwards the query to a LBS service provider. The LBS service provider generates a response and gives it to the network operator or wireless service provider, which forwards it back to the customer or user. The basic assumption in the literature is that customers or users will trust their network operator or wireless service provider for privacy protection, whereas location-based service providers are not trusted for such a purpose. We also assume the customers or users will send position information to a network operator with very high accuracy. Location information can be obtained either on the client itself (e.g., GPS) or by wireless service providers, for example through triangulation of the wireless signal. The location information is periodically transmitted to the network operator through a cellular or wireless network, and such information can then be sent to the location service provider either upon a query from the user, or periodically transmitted. Finally, we assume that the network operator will anonymize any user message before forwarding it to the LBS service provider. That is, all identifying information, such as names or mobile phone numbers, will be removed from the location service request received by the LBS provider.

Location cloaking methods imply that inaccuracy must be introduced into the location information sent to a service provider, and the location-based service must be provided

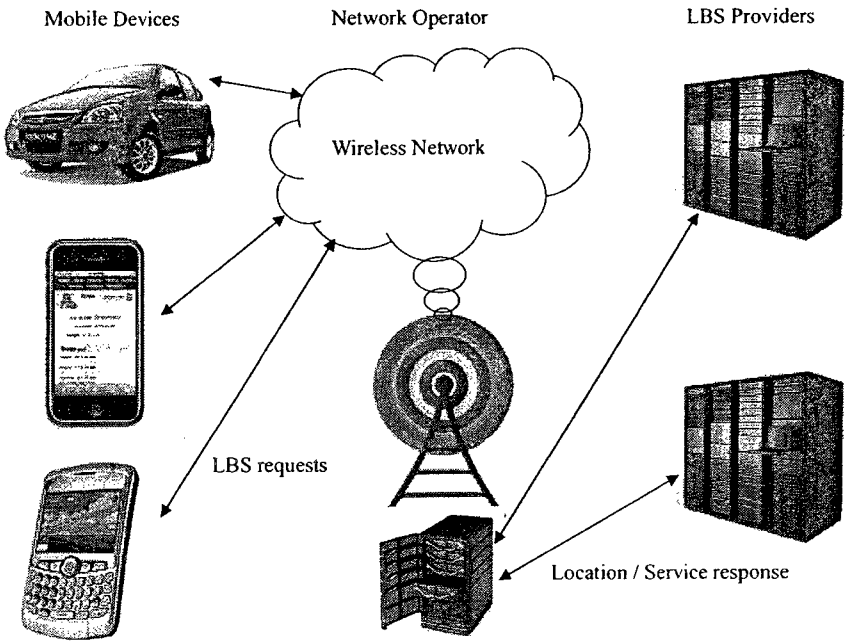

Figure 1: Architecture of LBS

using such inaccurate location information. We show this to be a reasonable assumption by considering several scenarios of location-based services. First, modern vehicles usually carry sensors that determine weather and road conditions. Therefore, instead of deploying expensive fixed sensors on highways, highway operators can obtain similar information by aggregating data collected from the in-vehicle sensors. For example, a rain sensor in a windshield wiper may detect rainfall and traction control sensors may detect icy road condition. Clearly, we do not need very accurate location information in this case and a road segment can be a good enough resolution. Also, temporal accuracy is not strictly necessary since most conditions do not change very quickly. Second, cars equipped with crash sensors or airbags may detect and report accidents. Such information can be used for traffic control statistics about accident risk at certain intersections or locations. Clearly, temporal accuracy is not at all important in this case, but location information must be accurate enough. Finally, for drivers requesting information such as area maps or nearby hotels,

the location information may be needed with varying accuracy requirements depending on specific services while the temporal accuracy requirement is much higher than previous cases.

1.2 Privacy Threat of LBS

As we have mentioned, the LBS service provider is not trusted in terms of users' privacy. This could be due to concerns over the fact that a LBS service provider may employ location information to profile the users and later push advertisements to the user. Also, due to the lack of direct binding between users and LBS service providers, the former may not be aware of the latter's privacy policies. Finally, the least privilege principle dictates that if a LBS service does not require the precise location information of a user, then the service provider should not be given accesses to such information. Therefore, the precise location information sent by users' devices should be cloaked before it is forwarded to the LBS service provider. Here we should distinguish between two classes of privacy threats. That is, communication privacy threats, which mean that adversaries on the network and LBS providers may attempt to decide who has sent an anonymous message. This thesis will focus on another type of issues, that is, an adversary may be able to reidentify the originator of an anonymous message using the location information within that message together with information gathered through out of bound channels, such as prior knowledge about a user, observations, or facts from the Internet.

In particular, three example scenarios have been identified in [21]:

- An adversary may correlate a location to a user if the location corresponds to an address that exclusively belongs to that user, and then the adversary will learn that the user sent the anonymous message. For example, the user could be the owner of a suburban house and he/she sends a message from that house; the coordinates of

that address can be correlated to the residence through a publicly available database of geocoded postal addresses. The user's identity can then be revealed through an address lookup in phone or property listings.

- If an adversary happens to know that a known user was at a particular location at a certain time, such as if the user has somehow revealed his/her identities in a previous message but now wants to send an anonymous message, then the message can be linked to the user using the location information.

- If an adversary can determine a sequence of locations all belong to the same users, then he/she can learn that user's route. If any of those messages reveals the user's identity, such as in the previous two scenarios, then the adversary can correlate the user to his/her route, and to obtain further information.

It is worth noting that location information may not directly violate users' privacy, but more sensitive information contained in the otherwise anonymous messages may be correlated to a user's identity through the location information.

1.3 Location Cloaking

To prevent the re-identification attack described above, location information needs to be cloaked before it is sent to the LBS provider. The original location cloaking method is proposed in [21]. The key idea is that a given degree of anonymity can be achieved by decreasing the accuracy of revealed spatial or temporal data. The algorithm simply chooses a sufficiently larger area, instead of the precise location coordinates, so that enough other subjects occupy the same area to prevent adversaries from determining the message originator. The desired degree of anonymity is specified by a parameter k, namely, the minimum acceptable anonymity degree. The algorithm is implemented in an anonymizing server

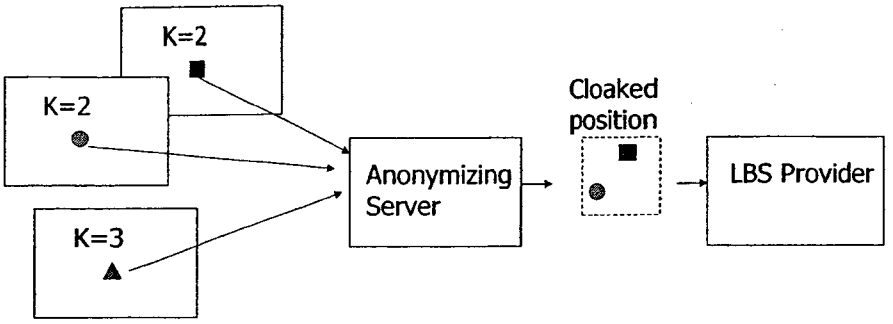

Figure 2: Location Cloaking

hosted at the network operator, as illustrated in Figure 2. It takes the current position of the requester and the current positions of other users in the same area. The algorithm will then subdivide the area around the requester's position until the number of users in the area falls below the constraint k. In the figure, the users represented by the rectangle and circle are cloaked together to satisfy both users' requirements, whereas the third user's requirement cannot be satisfied (suppose all three users send messages at the same time). An orthogonal approach is temporal cloaking, which reveals more accurate spatial location while reducing the temporal accuracy of messages. That is, we delay the request until $k - 1$ other users have traversed the same area occupied by the requestor. The request's time stamp is then modified to be this whole time interval with k users visiting this area, so adversaries cannot determine who sends the message.

In a later work, each user's constraints over the acceptable loss of accuracy in cloaked location, and hence in the quality of LBS service, is considered during location cloaking [13]. As illustrated in Figure 3, the solid-line boxes represent the maximum degree of cloaking that can be tolerated, whereas the dash-line boxes represent cloaking boxes. A simple observation can be made, that is, two users cannot be cloaked together while satisfying both users' tolerance levels, unless each user is inside the other's tolerance-level box.

More generally, a constraint graph is used to model multiple users' constraint boxes

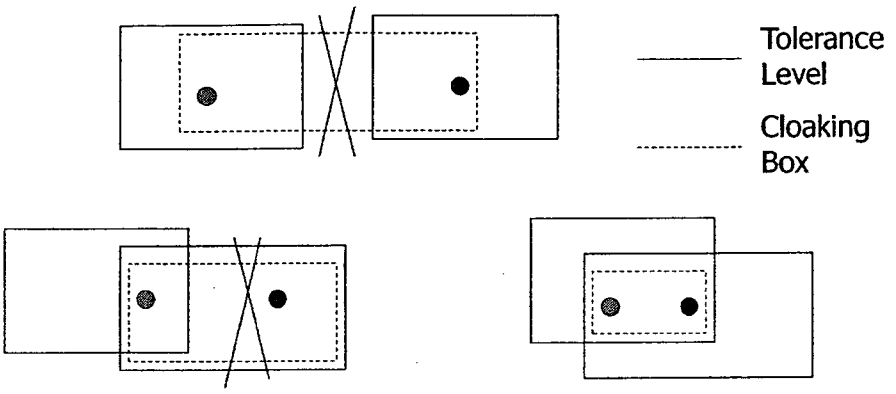

Figure 3: Location Cloaking with Constraints

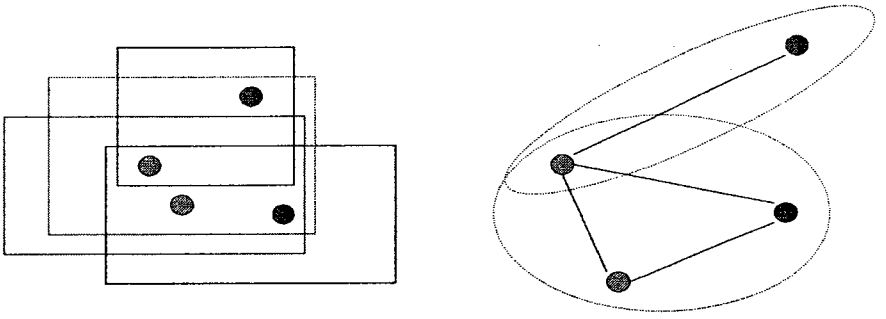

Figure 4: Clique Cloak Theorem

and relative locations in [13]. More precisely, the constraint graph is a simple undirected graph, with each user as a node and an edge connecting each pair of nodes that can be cloaked together based on aforementioned result. The so-called Clique-Cloak theorem is then derived to state that only nodes inside a clique (that is, a fully connected sub-graph) can be cloaked together, as illustrated in Figure 4. Notice that this does not necessarily mean the cloaking will be successful since each user may have his/her own value of k, which may not be satisfied by the cliques that user's node is in.

1.4 Simulatable Auditing

Our study of the location privacy auditing issue is inspired by the simulatable auditing problem in a different context, that is, statistical databases. In a statistical database, unlike in operational databases, users are allowed to ask for statistics, instead of individual values, so the objective is to prevent adversaries from inferring the latter from the former.

Let $X = \{X_1, X_2, \ldots, X_n\}$ be a simplified statistical database with $X_i \in R$ where $1 \leq i \leq n$ and R denotes the real numbers. The vector $x = (x_1, x_2, \ldots, x_n)$ denotes a database state. All queries over X take the form $q : R^n \rightarrow R$. The query auditing problem means that given a collection of queries $q_1, q_2, \ldots, q_{T-1}$ has already been answered with the answers $a_1, a_2, \ldots, a_{T-1}$, where each a_i can either be the true answer or "denied"; given a new query q_T, we must either give the true answer when none of the x_i's can be determined, or give "denied" as the answer, otherwise.

A naive strategy is to deny q_T only when needed, that is, when at least one of the a_i's can be decided. Surprisingly, this natural approach is actually not safe [28]. To illustrate this, consider the following example. Consider a database consisting of four integer variables $x = (x_1, x_2, x_3, x_4)$ and suppose the first query $q_1 : max(x_1, x_2)$ has already been answered with answer $a_1 = 5$. Suppose now the second query is $q_2 : max(x_2, x_3, x_4)$. Based on the above strategy, if the true answer happens to be $a_2 \geq 5$ then the true answer should be given because none of the variables can be uniquely determined. On the other hand, if the true answer is $a_2 < 5$ then we should deny q_2 because otherwise the value of x_1 can be inferred. Unfortunately, if q_2 gets denied, an adversary can immediately determine that $x_1 = 5$ because the only reason for the denial of q_2 under such strategies is that $a_2 < 5$, which means that $x_1 = 5$.

The *simulatable auditing* model [28] prevents the above attack by taking a different strategy. That is, whether to answer a newly posed query is determined based on only the knowledge that an adversary has already obtained, which include all answered queries and

their answers. In general, the simulatable auditing strategy can be stated as the following. Given a set $\mathcal{X}$ of all possible database states that are consistent with previously answered queries $(q_1, q_2, \ldots, q_{T-1})$ and their answers $(a_1, a_2, \ldots, a_{T-1})$, a new query q_T will be denied if only there exists at least one $x \in \mathcal{X}$, which is not necessarily the true database state, under which answering q_T is not safe. If we apply this new strategy to the above example, q_2 will always be denied no matter what possible answer a_2 is. Therefore, no inference is possible.

Chapter 2

Literature Review

The privacy issue of LBS has drawn significant attentions lately. Many methods have been developed for protecting the privacy information in LBS. The k-anonymity model-based temporal and spatial cloaking method [21] has been widely adopted as a baseline solution for providing privacy protection in LBS. In their algorithms, they use a trusted anonymity server to hide users' location information with $k-1$ other users. In their algorithms, when the resolution of cloaking area is too coarse to have a good LBS quality service, the temporal cloaking, which will delay a user's service request, will be applied. However, a real-time LBS service cannot tolerate any delay of service requests such as in real-time traffic reporting. Although there are some new cloaking methods that have been proposed recently [7, 31, 32, 43, 44, 50], the basic idea is still adding a third trusted party server to cloak the query before a request for LBS is sent to a provider, the location data in the request is replaced with a cloaked version, which may match at least k mobile devices. An adversary will not be able to determine from the cloaked location which of the k users has originated that request.

In order to accommodate different privacy requirements of different users, a personalized anonymization model is introduced in [13]. The authors propose a customizable

framework and provide the Clique-Cloak algorithm to support the framework. Each mobile node can specify its preferred minimum level of anonymity and the maximum temporal and spatial cloaking it is willing to tolerate in requesting for LBS; however, the algorithm implies high computation overhead and is mostly effective when the anonymity level that users ask is small. Although TTP(Trusted Third Party) based approach has been widely accepted, the problem of trust between mobile users and a LBS server just moves to user and third party entities. The difference is only that the third trust party can be well-known and the risk of trusting a dishonest entity is reduced. However, in many cases many users would rather like to trust nobody at all. Therefore, recently, there are also other privacy protecting methods [8, 15, 18, 24, 41, 42, 45, 51] for location based services that do not require the trusted third party to cloak the request query to the LBS server. Those methods mostly use cryptographic techniques to find out the nearest neighbors. Most of their algorithms are based on secure multiparty computation problem mentioned in [20]. Those methods are mostly collaboration-based, obfuscation-based, or PIR-based that can achieve the privacy protection to a certain degree as surveyed by Ghinita [17] and Abedelaziz [35].

A limitation of many cloaking methods is the high computing and processing cost. When using cloaked areas for LBS, specialized query processing algorithms may be needed, which may not be available in existing LBS servers. In order to overcome this problem, there is a recent SpaceTwist concept proposed by M.L Liu [48]. It neither belongs to the TTP-based or TTP-Free algorithms. In SpaceTwist, a mobile user will ask the nearest neighbors to send a fake location information called an anchor instead of sending its own location information to get the LBS. As mentioned above, the TTP based approach has obvious limitations compared to TTP free approaches, but a TTP based approach is easier and cheaper to be implemented in a LBS system. Our proposed cloaking algorithm is also based on a TTP based approach. Nishkam et al. has stated a similar idea as the inference

problem [37] but the solution is different since it is based on cryptography. They introduced a framework that ask users to send the code that include all sensitive information to the trust server, and then the trust server reads the code and sends back the results. Cooper and Morris [10]proposed a new concept to keep privacy of location information, which binds privacy rules with location information based on the web.

A special class of LBS queries, the Location-based Range Query(LRQ) is studied in [26]. A user issuing an LRQ is only interested in objects within a fixed distance from the current location. The solution for privacy preserving is to use an imprecise version of the query, namely, ILRQ [3, 26]. The inaccuracy is modeled as probabilistic guarantees to the answers indicating the degree of confidence in those answers. The shortcoming of centralized anonymizer is pointed out in [9]. Such anonymizer becomes a bottleneck in handling all queries, and is also a single point of failure. While the peer-to-peer spatial cloaking algorithm also cannot guarantee the success of cloaking when clients are sparsely distributed, the privacy exposure may happen when the requester tries to be in the center of the cloaking region. A distributed protocol is used by mobile entities to self-organize into a fault-tolerant overlay network in [19]. This approach can guarantee the query anonymity even when location information is disclosed to the adversary; however, each user must maintain a complex data structure and follow communication protocols that may cause additional computation and communication cost that limit clients' capabilities.

Other variations of the cloaking method include *Casper* [36] which consists of two components, the AS and the privacy-aware query process. The AS does the job as the anonymity server does, which blurs the user's exact location information into a cloaking area protected by desired protection requirement. The privacy-aware query processor is embedded inside the AS to deal with the cloaking area instead of exact location information. This cloaking algorithm uses quad-tree in a bottom-up fashion to find a spatial area

that meet all the cloaking requirements. The AS has to be known of the location information of mobile users in a fine spatial resolution, which will easily lead to a performance bottleneck when there is large volume of client's service request, and due to the limitation of the quad-tree structure, the resolution of the cloaking area is often larger than desired requirement that will for sure reduce the service quality.

There are also some work related to how to protect users' moving traces. Beresford and Stajano [1] proposed a concept called mix zone that is a spatial region in which a mobile user doesn not report its location information. When there are more than one users in the same mix zone, they exchange their pseudonyms, and use the new pseudonyms after they exit the mix zone. By doing that, an adversary will be difficult to find out a mobile user's moving trace and cannot make an estimation of their future locations. This algorithm relies on a set of predefined spatial zone for pseudonym exchanging. In their paper, they use the algorithm called path confusion algorithm which has been proposed by Hon and Gruster in [25]. This algorithm allows mobile users to exchange their pseudonyms with each other when they are close to each other. However, this algorithm has the obvious limitation of not being able to be applied to anonymous LBSs. Because in the algorithm, they have to report users' true location information to the service provider, it cannot support anonymous uses of LBS.

Marco Gruteser and Xuan Liu [23] proposed a method to hide sensitive areas (e.g., hospital and hotel) visited by users from adversaries. They classify areas into two categories, which is sensitive or insensitive areas. When user is in a sensitive area, its location information will be suppressed. otherwise, it reports its location. When it is moving close to a sensitive area, it will report an area with at least $k-1$ sensitive areas as its current location. This method cloaks $k-1$ sensitive areas rather than cloaking at leat $k-1$ users. By doing that, an adversary cannot know the sensitive area where a user goes. A simpler form of inference attacks is mentioned in [22]. Keeping anonymity of continuous moving

nodes is studied in [46,47], although the speed and direction are not assumed to be part of the location data. Their focus is on how to ensure enough uncertainty can be maintained when neighborhood changes in time, and how to find out the minimal cloaking area while still keeping anonymity level. We solve a different problem where inferences can be made using speed and direction information.

The work on finding nearest neighbors for continuous queries [4, 19, 49, 52] do not apply to the privacy issue of LBS but their techniques can certainly be integrated with our method for constructing cloaking boxes. M.gruteser and B.Hoh also stated an inference attack that an adversary can have an inference attack based on users' known speed and direction [34]; however, in our system, by using cloaking mechanism, an attacker can have an inference attack even by given the speed and direction information of cloaking box instead of individual user. Personalization and customization of cloaking method [13] has been broadly adopted in location-based services. It will increase the service quality of location-based service to the end users. We also proposed a cloaking method based on the traditional k-anonymity cloaking algorithm but at the mean time, the risk of privacy information leaking also arises. This issue has been addressed by Shin et al. [40]. In the paper, they pointed out that adversary can also identify the mobile users by looking for his/her customized security requirements, such as anonymity requirement, etc., while in our methods, the mobile user can personalize cloaking requirements without scarifying privacy.

Most previous researches on location privacy issues are based on snapshot queries. One example of the snapshot queries is that a mobile user looks for a well known POIs (Point of Interests). Because this kind of service is not real time, the anonymity server can easily use temporal cloaking when spatial cloaking cannot achieve the security requirements. Lars Kulik [30] states the problem of real time privacy protection in LBS. As he mentioned, how to maintain privacy security in the real-time LBS will be the greatest challengers for

the next generation of LBSs. And because of the high level of security level required by mobile users, their query cannot be passed by anonymity server. While this problem will not impact the performance of non real-time LBS because mobile users can wait until their requirements are satisfied by the anonymity server and their queries are passed to LBS server, it will significantly lower the service quality in real-time LBSs.

Audit systems have been used for information leakage control in databases. Given a set of data consisting of all information of users including sensitive information and a series of posted queries, the system denies the query that will expose the sensitive information and allows safe queries. There are two traditional auditing methods, which can be divided into online [12, 28, 38] and offline [5, 6, 27, 29] auditing methods. In the LBS system, we assume that attackers have knowledge of the cloaking algorithm for deciding the success of the cloaking result. Adversaries can trace the messenger originator by looking for their anonymity requirements and find out the cloaking results by using the same cloaking algorithms as the anonymous server uses. Therefore, we propose an online auditing algorithm that simulate the cloaking algorithm to protect such information leakage. It is to our best knowlege the first work to apply the simulatable concept to location based services.

Chapter 3

Inference Attack

In this chapter, we first describe examples of inference attacks using speed and heading direction. We then give a more formal model of the inference attack.

3.1 Motivating Examples

Consider again the classic scenario of privacy preserving LBS based on location cloaking [21]. There are three kinds of entities, namely, mobile devices owned by individual users, a wireless service provider, and third party LBS providers. The wireless service provider is considered as trusted in terms of user privacy, but the LBS providers are distrusted for this purpose. A cloaking server is hosted by the wireless service provider. Upon receiving an LBS query with location data from mobile devices, the cloaking server will replace the location with a less accurate version, such as a cloaking box (an axis-parallel box containing the actual coordinate of the device), such that this cloaked location will match at least k devices. The cloaked location is sent as part of an anonymized query to a LBS provider. An adversary at the LBS provider cannot determine which of the k devices has originated the query from the cloaked location, even if the adversary knows each user's precise location through out of bound channels, such as visual observations. This provides

the user with a certain degree of privacy protection.

However, the above cloaking technique becomes insufficient, if the LBS is based on a sequence of location data including the coordinate, speed, and heading direction of the same device at regular time intervals. The inherent geometric relationship between distance, speed, and direction will enable an adversary to deduce more accurate information than given cloaking boxes, such as that some coordinates inside the cloaking box are actually impossible for a device to reach. By removing such impossible coordinates from further consideration, the adversary may violate the privacy constraint that at least k devices should be included in the cloaking box. Figure 5 through Figure 8 illustrate such inference attacks under different assumptions about location data available to the LBS (and the adversary). In those figures, the black node represents a user device sending in an LBS query; the white and gray nodes represent other nearby devices; the boxes in solid line represent cloaking boxes for $k = 2$ at consecutive time points t_1, t_2, t_3.

Figure 5 depicts a simple case where the heading direction is in parallel to the x-axis and the speed is a fixed value v both of which are known to the adversary. The boxes in broken line represent the possible coordinates at t_2 and t_3 that an adversary can infer from the speed and time. The shaded boxes represent the adversary's final estimation of the message originator's location. Notice that at time t_3, the adversary infers possible coordinates using his/her inferred result (the shaded box) at time t_2, instead of the cloaking box at t_2. The estimation can thus be improved over time by intersecting many inferred results to reduce the amount of possible locations. Eventually, the adversary's estimation, as shown in the shaded box, may include only the black node, which means the privacy constraint $k = 2$ is now violated.

Inferences are not only possible when the exact speed and heading direction are available to adversaries. Figure 6 shows the case with a more relaxed assumption that the speed is only known to the adversary as in a range $[v_{min}, v_{max}]$ instead of a fixed value while the

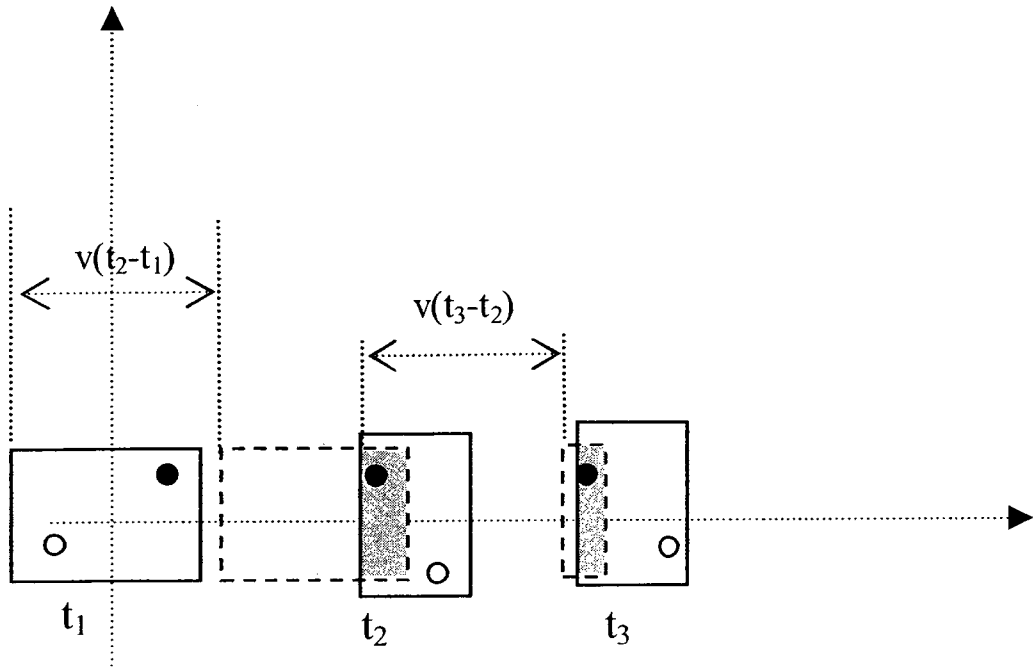

Figure 5: Inference Attack on Coordinate Cloaking Case 1

direction is still known to the adversary. It can be observed that starting from any coordinate in the cloaking box at t_1 and moving at any speed between v_{min} and v_{max}, a node will end up inside the box in broken line even though the exact speed is not known. Again, the adversary can eliminate impossible coordinates based on such inferences to reduce the size of the cloaking box at t_2, and eventually violate the privacy constraint since the deduced box of location includes only one node now.

Similarly, partially knowledge of the heading direction may also enable inference attacks. Figure 7 shows another case where the speed is known by adversaries as a fixed value v while the direction is only known to be inside a range $[\theta_{min}, \theta_{max}]$. In this case, the possible coordinates at t_2 form a shape bound by four lines (corresponding to part of the cloaking box at t_1 moved at v in θ_{min} and θ_{max}, respectively) and two arcs. Again, the adversary can intersect this shape with the cloaking box to eliminate some impossible part of the latter from furthe consideration in order to obtain an inferred box with reduced area

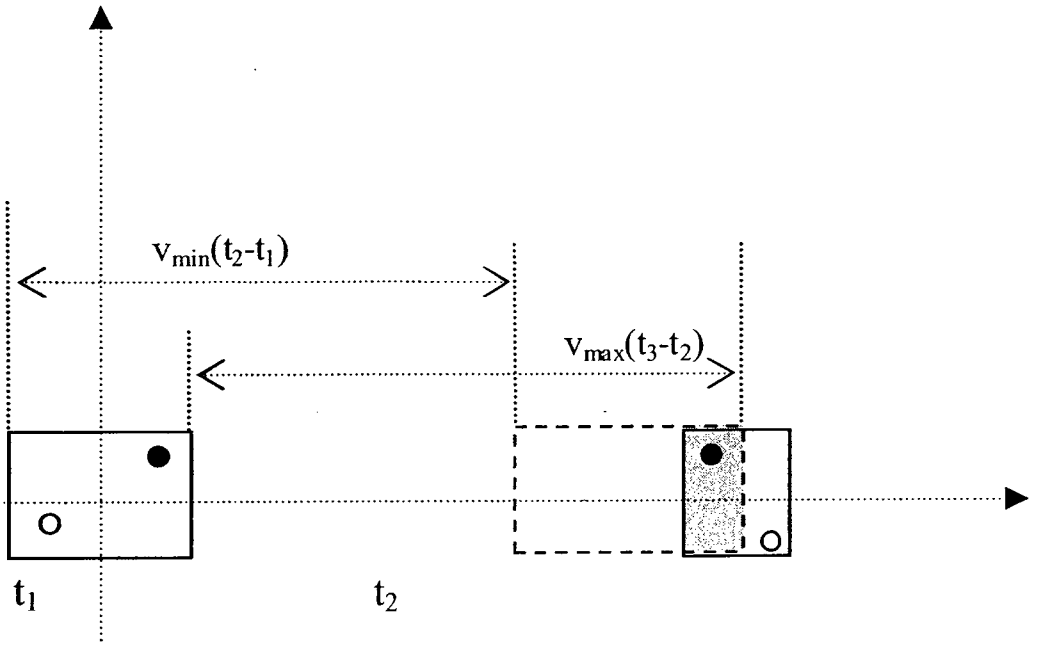

Figure 6: Inference Attack on Coordinate Cloaking Case 2

t_2. This may violate the privacy constraint as shown in this case that the deduced box of location includes only the message originator.

Finally, Figure 8 illustrates the following two interesting points.

- First, the upper half of the figure shows that a larger cloaking box at t_2 may solve the problem, since although the adversary can eliminate the white node from consideration, the gray node will still help to satisfy the privacy constraint $k = 2$. Notice that if the adversary is completely sure that the two cloaking boxes are about the same user, then he/she can exclude the gray node from consideration. This is another type of inferences that are already addressed in [46,47] and henace is not considered here.

- However, in the lower half of the figure, using a larger cloaking box to include the gray node does not help. In fact, since the cloaking box at t_2 does not affect the box in broken line (because possible coordinates at t_2 only depend on the cloaking box at t_1, speed, and direction), using a larger cloaking box at t_2 will not help in most

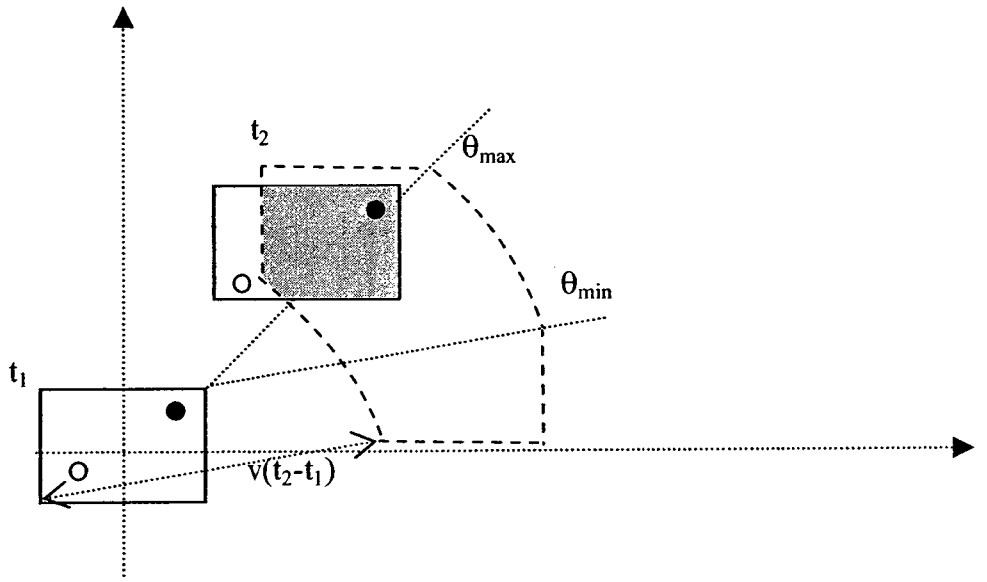

Figure 7: Inference Attack on Coordinate Cloaking Case 3

cases (this is true in all other three figures). Second, the lower half of the figure shows yet another case. The speed is unknown, so possible coordinates at t_2 form an infinite area between the two broken lines. Nevertheless, the privacy constraint is still violated.

3.2 The Model

In the following, we more precisely model the inference attacks illustrated in the previous section to facilitate further discussions.

Definition 1 *Given a collection of mobile devices D,*

- *We denote the coordinate, speed, and heading direction of a device $d \in D$ at time t as $c(d,t)$, $v(d,t)$, and $\theta(d,t)$, where both $v(d,t)$ and $\theta(d,t)$ can be in the form of a precise value, a range of possible values, or unknown.*

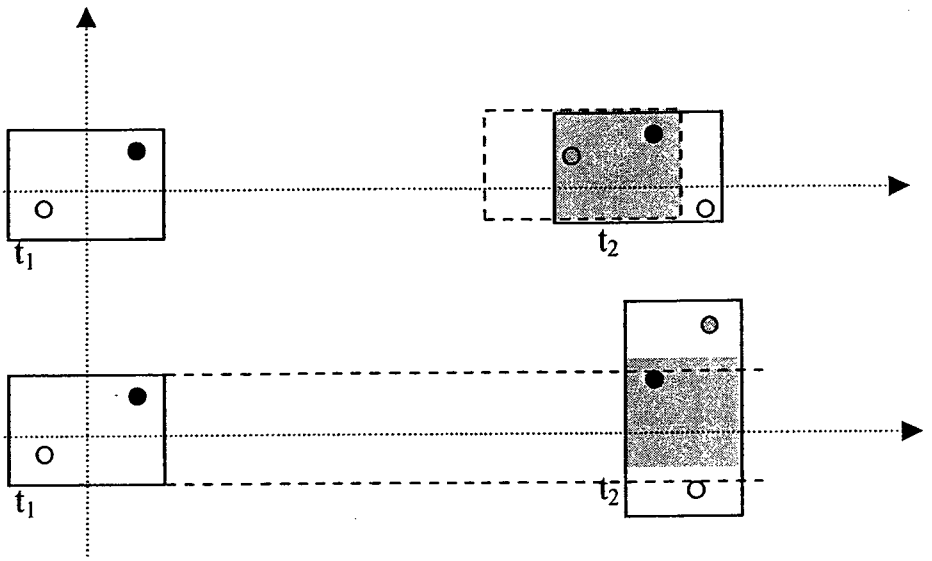

Figure 8: Inference Attack on Coordinate Cloaking Case 4

- *A* coordinate cloaking algorithm *takes as input an anonymity set size k, a time t, a node $d \in D$, a set of coordinates $\{c(d', t) \mid d' \in D\}$, and outputs an axis-parallel box that includes d, namely, the* cloaking box $cb(d, t)$ *(shown as solid boxes in Figure 1).*

- *Given a sequence of cloaking box, speed, and direction as $S = \{cb(d, t), v(d, t), \theta(d, t) \mid d \in D, t = t_1, t_2, \ldots, t_{n-1}\}$, we say c is a possible coordinate for d at time t_n, if assuming d is located at c does not cause a conflict with S. We say the collection of all possible coordinates for d at t_n as the* inference box $ib(d, t_n)$ *(shown as boxes in broken line in Figure 1). Notice that the word* box *is abused here since both the inference box and inferred box can potentially be any geometric area other than a box.*

- *We call $ib(d, t_n) \cap cb(d, t_n)$ the* inferred box *(shown as shaded areas in Figure 1). An inference attack happens if the inferred box includes less than k devices (k is the given anonymity set size).*

From above examples, we can make two observations as follows.

- $ib(d, t_n)$ is equal to the collection of coordinates that do not cause a conflict with $v(d, t_{n-1})$, $\theta(d, t_{n-1})$, and $ib(d, t_{n-1}) \cap cb(d, t_{n-1})$.
- The number of nodes in $ib(d, t_n) \cap cb(d, t_n)$ is no greater than that in $ib(d, t_n)$, and the latter is independent of the choice of $cb(d, t_n)$.

The implications of the first observation is that from now on we can focus on two consecutive time points, instead of a sequence of them (more details will be given in next section). The second observation basically announces that the inference attack cannot be addressed by increasing the degree of cloaking in existing algorithms. We are thus motivated to devise new cloaking methods.

Chapter 4

Inference-Free Cloaking

In this chapter, we first describe the basic speed and direction cloaking method for consecutive queries, and we then address several related issues.

4.1 Basic Cloaking Method

To prevent the aforementioned inference attack, it is necessary to cloak not only coordinates but also speed and/or direction. Examples given in the previous chapter have shown that cloaking coordinates only is not sufficient in all cases. In addition, cloaking either speed or direction alone is generally also insufficient. In Figure 8, we have shown that an inference is possible even when the speed is completely unknown (which is the extreme case of cloaking speed only). Figure 7 can be easily modified to show that even an unknown direction may still enable an inference attack (by maximizing the range of direction). Therefore, both direction and speed need to be cloaked. The question is how much we should cloak them. Clearly, an inference attack will no longer be possible if the inference box completely covers the cloaking box (or, the inferred box is the same as the cloaking box). More precisely, we define the concept of inference-free cloaking as follows.

Definition 2 *We say a cloaking algorithm is* inference-free *if it ensures that* $cb(d,t) \subseteq ib(d,t)$ *holds for any device* $d \in D$ *and any time* t.

There apparently exist many different solutions to an inference-free algorithm depending on the precedence of cloaking among speed, direction, and coordinates. In this thesis, we shall focus on the following approach, that is, to keep the cloaking boxes produced by a given coordinate cloaking algorithm, such as those in [13,21] unchanged, while cloaking the speed and direction. In another word, we give the coordinates a higher priority in terms of cloaking. The advantage of this approach is that when an LBS does not need direction or speed information, the cloaking server can simply send only the cloaking box (while suppressing the cloaked speed and direction) to the LBS provider (however, this by no means prevents other possible approaches from having their meaningful applications. For example, in some applications the accuracy of speed, or direction, or both could be more important than that of coordinates in which case the priority of cloaking should certainly be given to the former).

Figure 9 illustrates the situation where the black node needs to be cloaked at time t_n (the observations given in the previous chapter allow us to consider only two time points). As depicted in the left figure, we know the coordinate, speed, and direction of all three nodes at t_n. Moreover, we can also compute the inferred box at time t_n, shown as the shaded area (for the case of $n = 1$ this will be the cloaking box itself). Instead of sending the actual speed v_n and direction θ_n to the LBS provider, the cloaking server needs to send the cloaked version, $[v_{min}, v_{max}]$ and $[\theta_{min}, \theta_{max}]$, respectively. The objective of such cloaking is that at time t_{n+1}, as shown in the right figure, the inference box (in broken lines) will completely cover the cloaking box (in solid lines), which results in an inference box that is inference free.

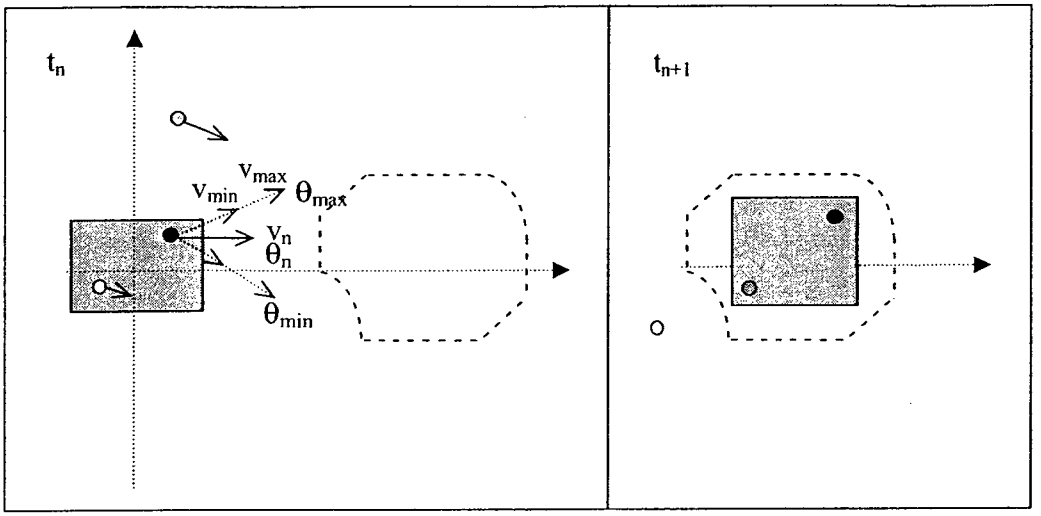

Figure 9: Cloaking Speed and Direction

However, the cloaking server must compute the amount of cloaking in speed and direction that is sufficient for the resultant inference box at the next time point to be inference-free, before we actually arrive at the next time point. That is, the cloaking must be determined in advance. For this reason, we are facing the following problem:

- On one hand, we need to compute and send $[v_{min}, v_{max}]$ and $[\theta_{min}, \theta_{max}]$ to the LBS provider at time t_n. Once these are known to an adversary, he/she can compute the inference box at time t_{n+1} (in broken lines in the left figure). Such knowledge of the adversary will not be reversible at a future time, which means the cloaked speed and direction can no longer be changed after t_n.

- On the other hand, at time t_n, we do not yet know the cloaking box at time t_{n+1} since it depends on future locations of nodes. This creates a difficulty in achieving the objective that the inference box at t_{n+1} (which must be decided at t_n and cannot be changed later on) completely covers the cloaking box at t_{n+1} (which is still unknown at t_n).

To address the above issue, we must estimate the missing information about t_{n+1} at t_n. More precisely, we estimate the cloaking box at t_{n+1} using the location information

of all nodes at t_n. Notice that it may seem to be a viable solution to simply estimate the cloaking box at t_{n+1} using the cloaking box, speed, and direction at t_n, regardless of other nodes' location. However, this approach will not provide a good estimation since most coordinate cloaking algorithms are based on the nearest neighbors, which may change in time depending on other nodes' movements relative to this node. As shown in Figure 9, the black node is cloaked together with the white node at t_n, but with the gray node at t_{n+1}.

Therefore, we choose to first estimate future locations of all nodes based on their current locations, and then compute the future cloaking box based on the estimated locations. Finally, we compute the current cloaked speed and direction using the estimated future cloaking box. This process is more precisely described in the following algorithm.

Algorithm *Inference-Free Cloaking*
Input: A set of nodes D and $\{c(d,t_n), v(d,t_n), \theta(d,t_n) \mid d \in D\}$,
a node $d_0 \in D$, $ib(d_0,t_n) \cap cb(d_0,t_n)$, and anonymity requirement k.
Output: The cloaking result $v_{min}, v_{max}, \theta_{min}, \theta_{max}$.
Method:
1 For each $d \in D$
2 Estimate $c(d,t_{n+1})$ from $c(d,t_n)$, $v(d,t_n)$, and $\theta(d,t_n)$;
3 Compute $cb(d_0,t_{n+1})$ from $\{c(d,t_{n+1}) \mid d \in D\}$;
4 Compute intervals $[v_{min}, v_{max}]$ and $[\theta_{min}, \theta_{max}]$ to have
5 the smallest set $ib(d_0,t_{n+1}) \supseteq cb(d_0,t_{n+1})$;
6 Return $v_{min}, v_{max}, \theta_{min}, \theta_{max}$;

4.2 The Amount of Cloaking

Unlike in coordinate-based cloaking where the inferred box is uniquely determined once a cloaking box is given, with cloaking of speed and direction, the inferred box must be determined in a different way since it will also depend on the cloaked speed and direction. To clarify, we compare the required steps for determining the inference box in both cases below.

First, with coordinate-based cloaking, the required steps are the following.

1. At time t_n, the device gives the current coordinates $c(d, t_n)$.
2. At time t_n, the cloaking server publishes the current cloaking box $cb(d, t_n)$.
3. At time t_n , the adversary can compute the future inference box $ib(d, t_{n+1})$ based on speed and direction.
4. At time t_{n+1}, the device gives the current coordinates $c(d, t_{n+1})$
5. At time t_{n+1}, the cloaking server publishes the current cloaking box $cb(d, t_{n+1})$.
6. At time t_{n+1}, the adversary computes the current inferred box $ib(d, t_{n+1}) \cap cb(d, t_{n+1})$.

Next, with speed and direction-based cloaking, the steps required for determining the inferred box are the following.

1. At time t_n, the device gives the current coordinates $c(d, t_n), v(d, t_n), \theta(d, t_n)$.
2. At time t_n, the cloaking server publishes the current cloaking box $cb(d, t_n)$.
3. At time t_n, the cloaking server estimates the future cloaking box $cb'(d, t_{n+1})$.
4. At time t_n, the cloaking server can compute and publish the cloaked speed and direction based on $cb'(d, t_{n+1})$, $v(d, t_n)$, and $\theta(d, t_n)$ such that the corresponding $ib(d, t_{n+1})$ satisfying $cb'(d, t_{n+1}) \subseteq ib(d, t_{n+1})$ (as detailed later in this section).
5. At time t_n, the adversary can compute the future inference box $ib(d, t_{n+1})$ based on $cb(d, t_n)$ and the cloaked speed and direction.
6. At time t_{n+1}, the device gives the current coordinates $c(d, t_{n+1}), v(d, t_{n+1}), \theta(d, t_{n+1})$.
7. At time t_{n+1}, the cloaking server publishes the current cloaking box $cb(d, t_{n+1})$.

8. At time t_{n+1}, the adversary computes the current inferred box $ib(d, t_{n+1}) \cap cb(d, t_{n+1})$.

With speed and direction-based cloaking, our objective is to ensure that at step 5, the cloaked speed and direction should satisfy that the inference box $ib(d, t_{n+1})$ computed by adversary at step 6 should completely cover $cb'(d, t_{n+1})$ (and hopefully $cb(d, t_{n+1})$ later at t_{n+1}, if $cb'(d, t_{n+1}) = cb(d, t_{n+1})$ happens to be true). However, the inference box $ib(d, t_{n+1})$ also depends on the cloaked speed and direction. Such an inter-dependency makes it more challenging to compute the required amount of cloaking in speed and direction.

We now approach this issue from another aspect. First we make a few observations as follows.

- As a general principle of location cloaking (or more generally, that of data generalization), we will provide less precise location information but we would never lie, that is, about the exact location. That is, any cloaked location information must enclose the actual location, including the speed and direction. Therefore, the inference box $ib(d, t_{n+1})$, which is computed based on the cloaked speed and direction, must enclose another inference box $ib'(d, t_{n+1})$ computed based on the actual speed and direction.

- When we cloak the speed and direction to a speed range and direction range, the inference box $ib(d, t_{n+1})$, which is certainly no longer a box, can be regarded as a union of many copies of the inference box $ib'(d, t_{n+1})$ each of which is based on a different combination of speed and direction in the cloaked ranges.

- Since the inference box $ib(d, t_{n+1})$ needs only to enclose both the estimated cloaking box $cb'(d, t_{n+1})$ and the inference box $ib'(d, t_{n+1})$, we can now determine it more easily based on the union of the two.

Figure 10 illustrates the above observations with an example where the unfilled box $ABCD$ is the cloaking box of a node at time t_n; the dark grey box $A'B'C'D'$ is the estimated future cloaking box at $t_(n+1)$; the light grey box $MNXY$ is the inference box computed based on the actual speed and direction ($cb'(d, t_{n+1})$ in above discussions).

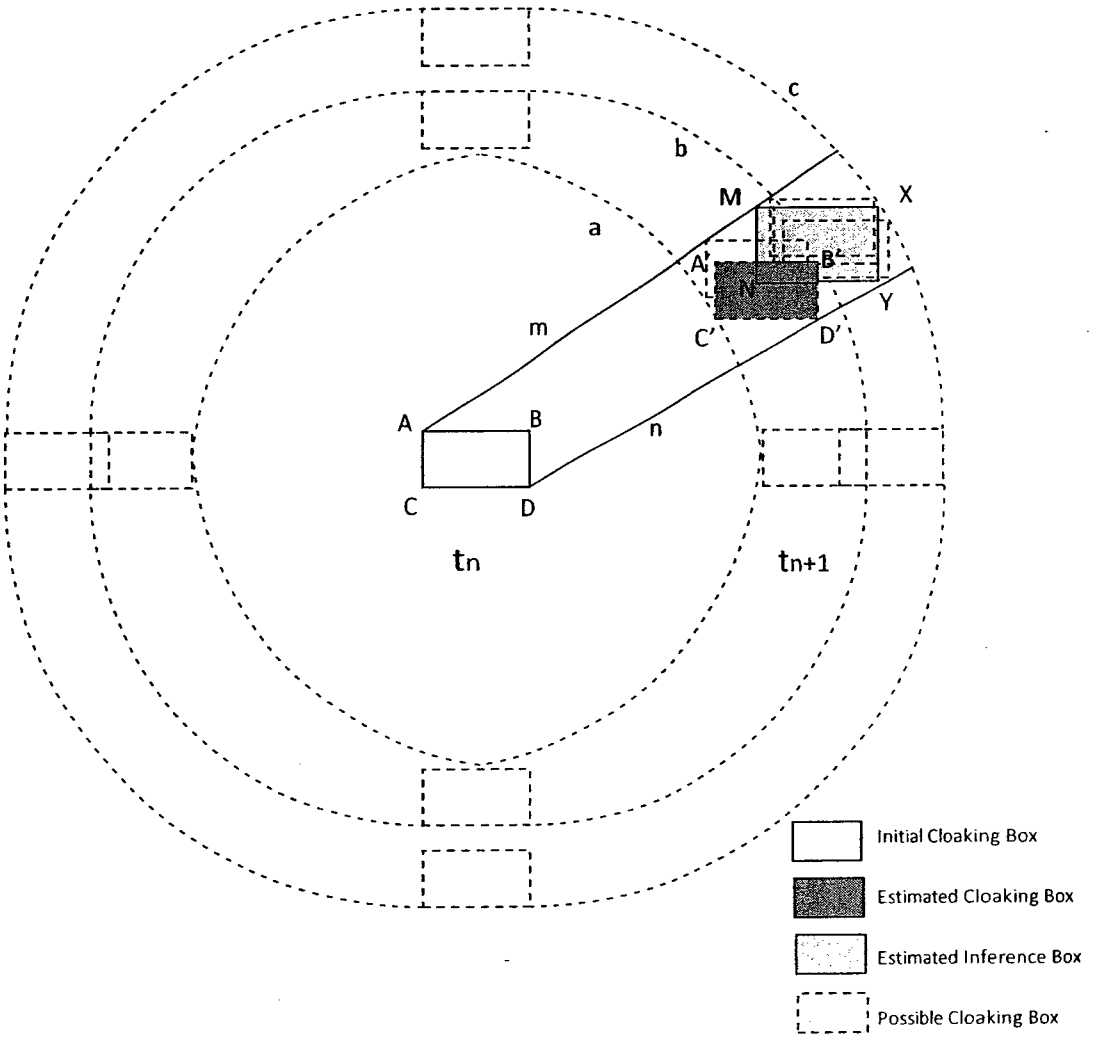

Figure 10: Cloaking Result

In this case, to find the minimal speed cloaking range that causes the inference box $ib(d, t_{n+1})$ (not shown in the figure) to fully cover both the dark and light grey boxes, we have arcs a and c where a is obtained from a circle with the point C as the center and CC'

as the radius; arc c is from a circle with the point B as the center and BX as the radius. Now, we can determine the required speed cloaking range by having the maximum speed as the length BX divided by $t_{n+1} - t_n$ and the minimum speed as CC' divided by $t_{n+1} - t_n$. Notice that we choose points B and C since all other points inside the box $ABCD$ can only reach the area enclosed by the arc a and c when they travel with any speed with the cloaked speed range.

Similarly, we can find the minimally required direction cloaking range. First, the dashed area indicates the inference box if the direction is not known. We can then find the two points M and D where the maximum cloaked direction is obtained from the line AM and the minimum cloaked direction from DD'. All other points will fall between those two lines if they travel along any direction between the cloaked direction range. Based on the cloaked speed and direction, an inference box can be computed, as illustrated in Figure 11.

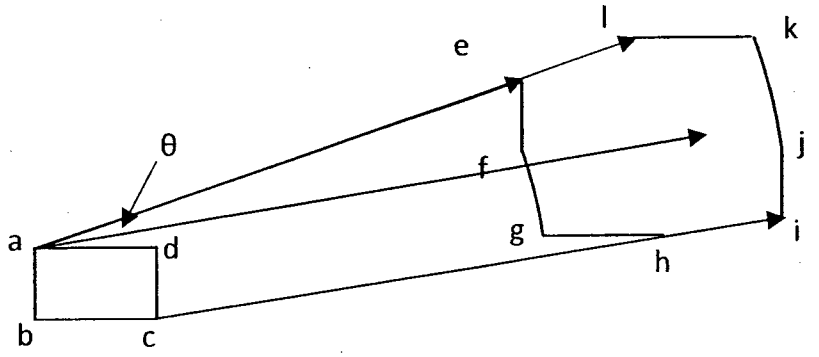

Figure 11: Cloaking Result

4.3 Other Issues

We now address several practical issues related to the previous speed and heading direction cloaking method.

4.3.1 Membership-Based Inference

Another type of inferences are possible based on memberships of a cloaking box. As shown in figure 12, the cloaking box at time $t0$ includes the black and white nodes. Assume the adversary know that the message is sent by one of these two nodes. In another word, either the black or white node is the message originator. At time $T1$, because the white node moves away from black node while grey node moves closer to the black node, the new cloaking box will include the black and grey nodes instead. Therefore, the adversary will know that at time $T1$ the same originator must still be either the black or grey node, that is, the grey node can be excluded from consideration through this membership-based inference. The message originator is thus exposed to the adversary.

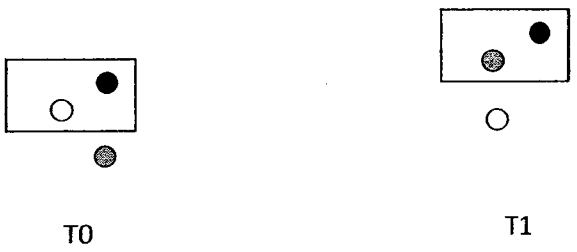

Figure 12: Membership-Based Inference

It would seem to be necessary to always include all the members inside a cloaking box at any subsequent time in order to prevent such inferences. However, this will certainly lead to a monotonically increasing cloaking box which eventually render the quality of LBS unusable for users. Fortunately, the above assumption about adversaries' knowledge about each user device's precise location is in practice too strong, and the knowledge about each device owner's identity is usually absent or limited [39].

As showed in Figure 13, initially when A and B are cloaked together, the probability of each device being the message originator is equal. At next time point, device B leaves and device C is cloaked with A. Assume that the adversary has no knowledge about who has left and who stays [11], then the possibility that device A is the message originator drops

to 25% since now A (or B, which the adversary does not know) and C are equally likely to be the originator. The adversary has no additional knowledge about who left and who stay. That is, the membership-based inferences are not possible unless if the adversary has complete knowledge about device owner's identity, which is not likely since all messages are anonymous. Another approach for reducing such a risk is to use a path confusion algorithm such that any adversary will not know who will be in a cloaking area [25].

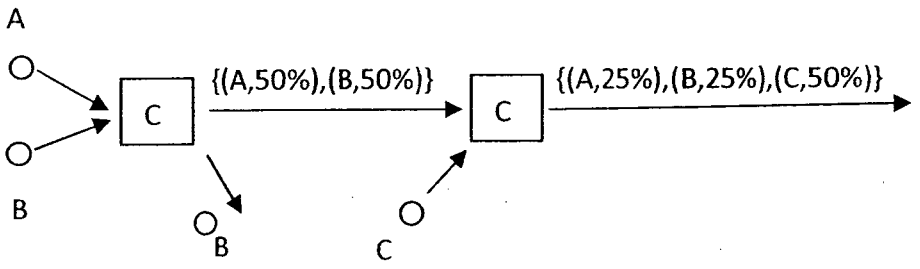

Figure 13: Probability of Being the Message Originator

4.3.2 Estimation Error Tolerance

The cloaking algorithm proposed in the previous section can prevent an inference attack in ideal situations where all information are accurate. However, in practice, an error may arise in many ways. The estimation of future locations of nodes may be inaccurate due to changes in speed and direction within given time intervals. Moreover, different nodes may send in their locations at different and varying time intervals, so even the current location of a node may not be available and must be estimated from a previously known location. The errors introduced by estimated locations may cause a privacy constraint to be violated if the actual inferred box is not fully covered by the computed one.

Our solution is to increase the level of cloaking based on estimated errors such that the cloaking result will remain valid even in the presence of such errors. An alternative solution is to simply increase the anonymity level k but we believe this will lead to a more significant

increase in the level of cloaking. Figure 14 illustrates our approach to the tolerance of estimated errors. The upper left figure shows the case where the cloaking algorithm does not consider any error. After the algorithm cloaks the speed and direction of the black node based on estimated locations, an inference box in the broken lines is known to the adversary. Then the upper right figure shows the actual future location of each node, which is different from the estimated ones. Since the gray node has moved further downward than estimated, the cloaking box must be enlarged to include it. However, the adversary already knows about the inference box, so the inferred box will still include just the black node itself, violating the privacy constraint.

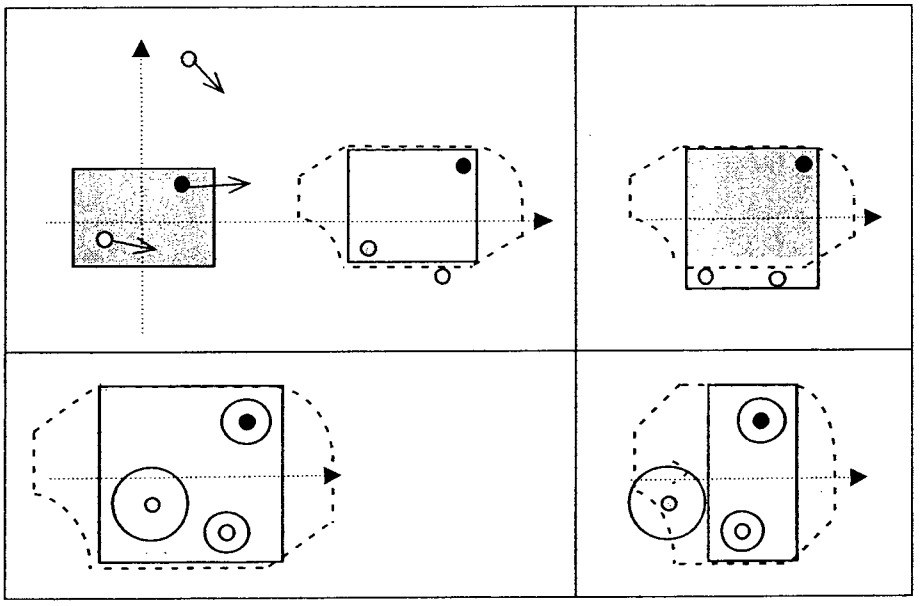

Figure 14: Tolerating Estimation Errors

For simplicity, we consider the case where estimation error is bound by a circle around each node (the circle will be replaced by a different shape when errors are estimated on each node's current location or its speed and direction). Starting from the estimated cloaking box in the upper left figure, we enlarge it by considering error bounds (the circles around nodes) to obtain the result shown in the lower left figure. The larger cloaking box will tolerate the

estimated errors. However, this also means a larger inference box, which in turn implies a higher degree of cloaking in speed and direction as the cost for tolerating errors. The lower right figure shows that the previous result is not optimal since the white node with a smaller error bound is actually a better choice for the cloaking. That is, we should directly estimate the cloaking box based on error bounds of all nodes, instead of enlarging a cloaking box computed from nodes without error bounds.

Chapter 5

Simulatable Auditing in Location Cloaking

In this chapter, we first study the possibility of extending the speed and direction cloaking method with customized constraints. We then show that additional inferences are possible when cloaking fails to satisfy such constraints. Finally, we discuss applying simulatable auditing to location cloaking to prevent such inferences.

5.1 Customized Location Cloaking

The speed and direction cloaking method introduced so far can address the inference attacks when the speed and direction can be freely cloaked as required. However, in reality this is usually not the case. The quality of the location-base service may be decreased to an unacceptable degree by forwarding cloaked speed and direction to the service provider. We thus consider cases where customized constraints are given about such cloaking. That is, similar to constraint boxes that must completely cover any cloaking box [13], users may indicate a range of speed and heading direction that must enclose any cloaked speed and direction.

Clearly, due to the inherent geometric relationship between speed, direction, and moving distance, it is not always possible to meet all the cloaking constraints over coordinates, speed, and direction. The cloaking algorithm must thus know how to tradeoff between these constraints and reach an acceptable solution. Moreover, even when all such constraints can be satisfied simultaneously, users may have preferences over which of the cloaking options should be given more priority. That is, some LBS services may benefit from a more precise cloaking box than a more precise direction, and vice versa. Therefore, the algorithm should take such preferences into account when computing the amount of cloaking.

Notice that we have concluded in the previous section that the amount of cloaking in speed and direction can be uniquely determined based on the estimated cloaking box. It may seem that no flexibility then exists in choosing to cloak speed or direction more. However, the cloaking box is not unique, and changing the shape of cloaking box (within its constraint box, which is not shown here) will enable us to give the cloaking of speed or direction more priority.

In Figure 15, we consider a case where different preferences over the cloaking of speed and direction may result in different cloaking solutions. The first case shows that without any predefined preferences, when the black node sends a request, it will be cloaked with the white node by the nearest neighbor-based cloaking algorithm. In the second case, the black node also originates a request but it asks to give direction cloaking higher priority. Therefore, the black node will be cloaked with the grey node in this case since by cloaking with the grey node, the result will lead to a smaller direction cloaking range (not shown). In the last case, the speed cloaking is given higher priority, and the black node will also be cloaked with the white node to yield a smaller speed cloaking range.

Table 5.1 shows an algorithm to support the above illustrated customized cloaking of speed and direction. The algorithm first apply the aforementioned Clique-Cloak algorithm [13] to future estimated location of nodes to find those nodes that can be cloaked

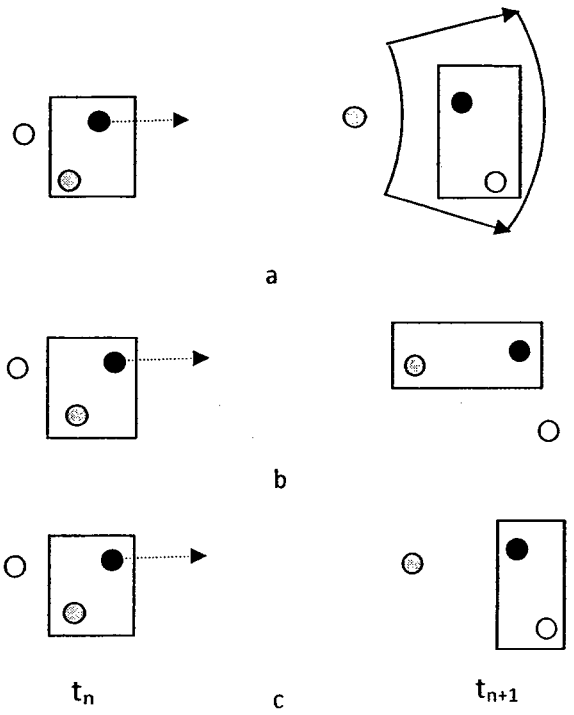

Figure 15: Customized Cloaking

together with the requesting one. It then finds $k-1$ neighbors among those nodes according to the given preference. Once the cloaking box is estimated, it applies the previous cloaking method to compute the amount of cloaking in speed and direction that is needed.

After getting the cloaking box, we then use the method we proposed in section 4.1 to finalize the speed and direction cloaking.

5.2 Simulatable Auditing

In previous work, such as [13,14], as well as in Algorithm *Customized Cloaking Algorithm*, the cloaking process will not always be successful. Existing solution is to employ temporal cloaking, that is, to simply delay the request for a while until the cloaking can be successful. Such a solution, however, cannot support a real time service that demands short delays. Also, if the mobile devices update their locations including the speed and direction to the

Algorithm *Customized Cloaking Algorithm*
Input: A set of nodes D, a message originator d and its anonymity requirement k and preferences over speed or direction cloaking.
Output: Cloaking result, or ϕ if cloaking fails.
Method:

```
1  Apply the Clique-Cloak algorithm to estimated locations at t_{n+1} to find all cliques in
       the constraint graph [13]
2  If the size of cliques including d is less than k
3    Return φ
4  If the preference is over speed cloaking
5    Search for k − 1 nodes in the same clique as d that will lead to the least
         amount of speed cloaking
6  Else
7    Search for k − 1 nodes in the same clique as d that will lead to the least
         amount of direction cloaking
8  If the search fails
9    Return φ
10 Estimate the cloaking box at t_{n+1} based on the k − 1 nodes
11 Apply Algorithm Inference-Free Cloaking to return the cloaked speed and direction
```

LBS service provider at a regular time interval, then the previous solution will not work, either.

When the cloaking server fails to cloak a requesting device, the LBS provider will not receive an updated location at the expected time point. An adversary at the LBS provider can then make an inference that the missing location update signals the device cannot be cloaked at the time to meet the customized constraints (which are assumed to be publicly known). Such extra information may lead to the breach of privacy requirements in many cases, as we shall show shortly.

In figure 16, the nodes stand for mobile devices and the correspond dash boxes are the cloaking constraint boxes of those nodes. The cloaking server therefore must cloak the location based on both the anonymity requirement k and the constraints, and only send successful results to the LBS provider. Based on the Clique-Cloak algorithm [13], the white circle node can be successfully cloaked with the black circle node in the spatial layout I.

However, in the spatial layouts II and III, the white circle node can no longer be cloaked at all according to the constraint boxes. In layout II, the white circle node is not in any other nodes' constraint boxes, and in layout III, although the white circle node and the black triangle node are in the same clique, the black triangle node's anonymity requirement k is equal 3, which means the clique cannot satisfy the requirement.

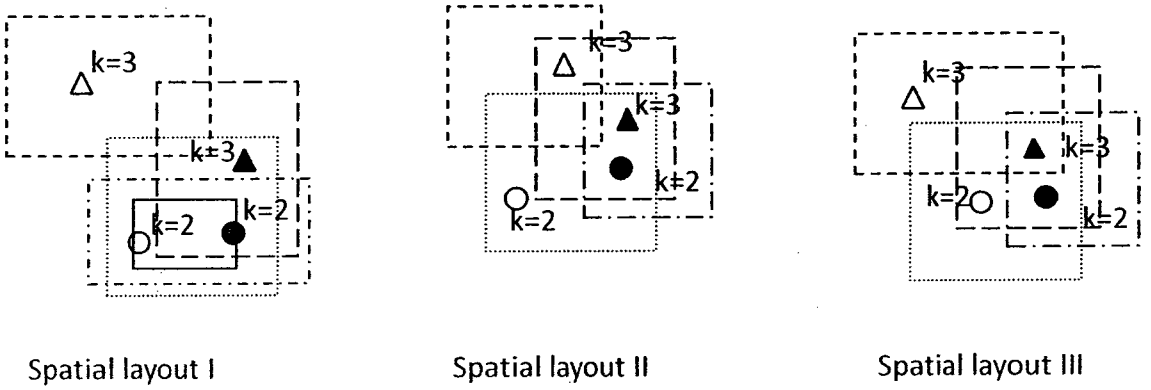

Figure 16: An Example of Clique-Cloak Algorithm

In Figure 16 spatial layout I, the white triangle node is sending the updated location to the server at a fixed time interval. When an adversary does not receive the updated location at any time point, he/she can infer that the location cannot be cloaked to meet the cloaking requirement. If he/she knows the locations of all nodes, he/she may find out that only the white triangle node and black triangle node cannot be cloaked at that time. The probability of each of them to be the message originator is $1/2$, which is greater than the probability $1/3$ implied by the anonymity requirements of the white triangle node, that is, the cloaking fails to protect its privacy.

The basic idea of simulatable auditing is to ensure that all nodes who will be denied of requests (more precisely, whose location information cannot be cloaked and will be withdrawn from the LBS provider) are still inside a group that satisfies all group members' anonymity requirements. We notice the following important points in applying simulatable auditing to location cloaking.

- All nodes that cannot be successfully cloaked will form the *denial group*. Unlike other nodes, nodes in denial group can be far way from each other, since this will not reduce the quality of LBS (no such service is provided to those nodes at that time point).

- Clearly, it is not guaranteed that there are enough nodes whose location cannot be successfully cloaked. Therefore, we may need to sacrifice some nodes whose location can be successfully cloaked by including them in the denial group.

- The denial group is formed based only on nodes' location, regardless whether they are sending any LBS requests at that particular time point. That is, as long as at least one of the member nodes cannot be cloaked when it sends a request (although it may not be the one actually sending requests), then all members' requests will be withdrawn from sending to the LBS provider. This is a key concept of simulatable auditing. To see the reason, assume we adopt a more relaxed policy of forming the denial group only when the requesting node's location cannot be cloaked. If there is only one such node while all other nodes' location can be successfully cloaked, then the adversary will be able to infer the former to be the requesting node, since the denial group will not be formed at all if any of others is the requesting node (he/she knows there is only one requesting node when seeing a single request).

In Figure 17, we illustrate two cases of forming a denial group using the constraint graph. In the auditing algorithm that follows, lines 4-6 correspond to the case I: The black node cannot be cloaked with the grey node since the grey node's anonymity requirement is 3. The algorithm will thus include the black node in the denial group because one of the nodes (grey node) is in the same clique as the black one. Because the maximum anonymity requirement is 3, which is more than the number of members in this group, all members in the group will be denied of requests (although not necessarily all of them are actually

sending a request). Line 7-17 of the algorithm correspond to case II. The black node with anonymity requirement of 2 forms no clique with other members (line 7). As line 8-11 describe, the algorithm will search for the nearest clique group that cannot be cloaked and group them together as the denial group. If the number of members in the group is greater of equal to the maximum anonymity requirement of the members then the grouping is successful. In case II, the black node will be grouped with the triangle and square nodes because the clique with those two nodes cannot be cloaked since the triangle node's anonymity requirement cannot be satisfied. Line 12-14 deal with the situation where the previous search fails. Then, the algorithm will search for the nearest clique to be grouped together to form the denial group.

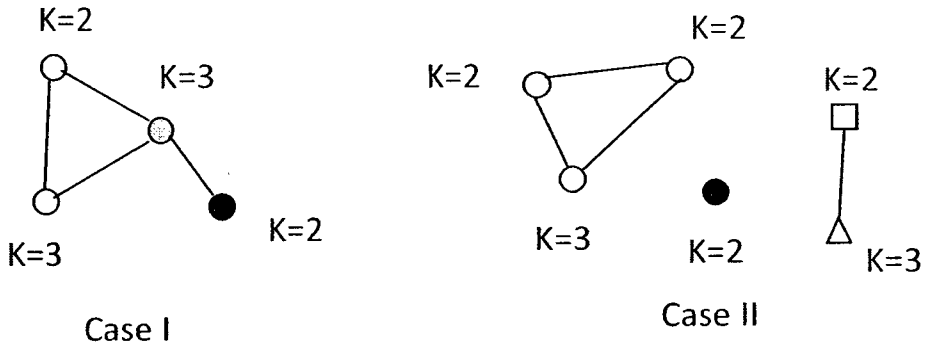

Figure 17: Forming a Denial Group

Algorithm *Audit System*
Input: Cloaking result from previous algorithms
Output: The denial group and its complement $G(D), G(S)$.
Method:

```
1   Call Clique-Cloak algorithm and place all cliques into G(S);
2   For each d ∈ D
3   If d cannot be cloaked;
4     Search for the cliques including d;
5     If there exists cliques that can satisfy all members' anonymity requirement;
6       Choose the smallest group to be G(D) and remove it from G(S);
7     If d is not inside any clique;
8       Search for the nearest clique that cannot be cloaked
9       Place the clique into G(D) and remove it from G(S);
10    Continue until the size of G(D) is no less than
11    the maximum anonymity requirement of all members;
12    If the above search fails;
13      Search for the smallest clique in G(S);
14      Merge the clique with G(D);
15    Continue until the size of G(D) is no less than the the maximum anonymity
                requirement of all members;
16    Return G(D), G(S);
```

Chapter 6

Simulation Results on Real World Map

We use the Network-Based Generator of Moving Objects [2] to generate random mobile devices and simulate their movements on the real city map of Oldenburg, Germany, a city about 15x15 km^2 (see figure 18). We generate several groups of mobile devices to study the effectiveness of our speed and direction cloaking algorithms in different density of moving objects. In this simulation, we select devices from some part of the city with all devices sending their service requests between a fixed time interval in a continuous manner. We maintain an estimation database to record all devices' estimated future locations computed based on the time span and the location information including speed and heading direction. The main objective is to measure the accuracy of cloaked location by applying our algorithms. We also conduct a comparison study between the speed and direction cloaking algorithm with and without customization constraints, and we also analyze the cloaking success rate. The terms used during further discussions is given in table 1. The following figures from figure 19 to figure 30 will show the simulation results obtained from the experiments.

Figure 19 and figure 20 show the speed and direction cloaking results with the size of the time interval changing from 1s to 4s. The results also show the cloaking results under

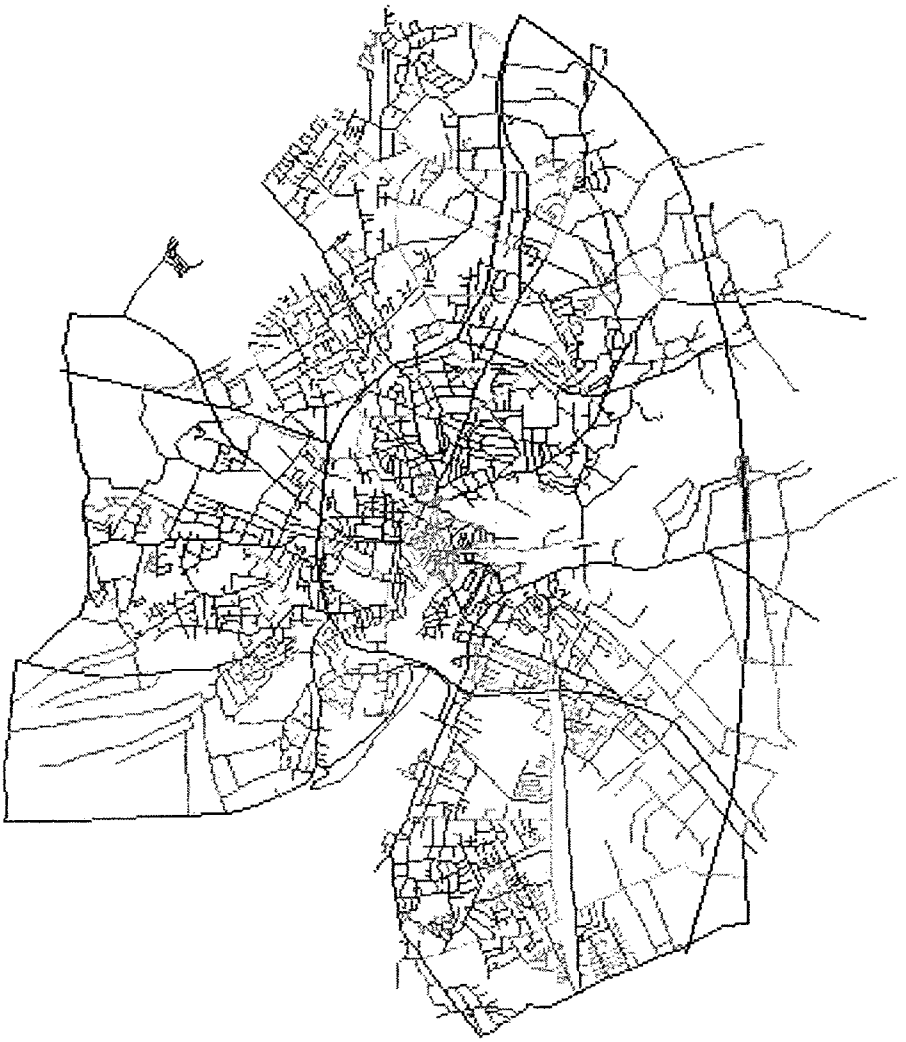

Figure 18: Oldenburg,Germany

different degrees of node density. From the figures, we can see both speed cloaking range and direction range decrease when mobile devices update their location information to the anonymity server at a big time interval(less frequently). When the device sends its LBS requests less frequently, the distance between the initial cloaking box and future cloaking box becomes larger. However, the absolute amount of cloaking will remain roughly the same. Therefore, the relative percentage of cloaking becomes smaller. We also find out the

Term	Meaning
C	The size of a constraint box
K	Anonymity requirement
Speed Range	(max_speed - min_speed)/real_speed
Direction Range	max_direction - min_direction
Success Rate	The rate of successful cloaking
Time Interval	The interval after which devices update their locations

Table 1: Terms Used in the Simulation

both speed and direction cloaking range become smaller when the density of devices increases. This is straightforward since when the density of devices increases, it will become possible to find a smaller cloaking box with enough nodes inside.

We also conduct a simulation on how the cloaking results change when devices' constraints change. We randomly choose 3000 nodes on the city map as a sample, with a fixed anonymity requirement and the size of time intervals. We cloak the nodes' speed and direction under different constraints. We made simulation with five constraint levels (C)(see Table 1) from 100 to 500. We find the accuracy of cloaked location decreases when the constraint is bigger (see Figure 21 and figure 22). This result is as expected since the constraint states the maximally tolerable amount of cloaking. The bigger constraint box is the easier to find enough neighbors to be cloaked together. Therefore, the success rate of cloaking will increase (see Figure 23).

We conduct a simulation to find out how the anonymity requirement may affect the cloaking result. In the simulation, we use 2000 and 3000 nodes with the same size of time intervals and constraints. As shown in Figure 24 and Figure 25, we find the speed and direction cloaking increase when the anonymity requirement becomes higher, which means when users desire better privacy protection the quality of location-based service will decrease. Also, as showed in Figure 26, when the k-anonymity level increases, the difficulty of finding $k-1$ neighbors increases so the success rate of cloaking drops. We again make the comparison between different density levels in the simulation. The result is

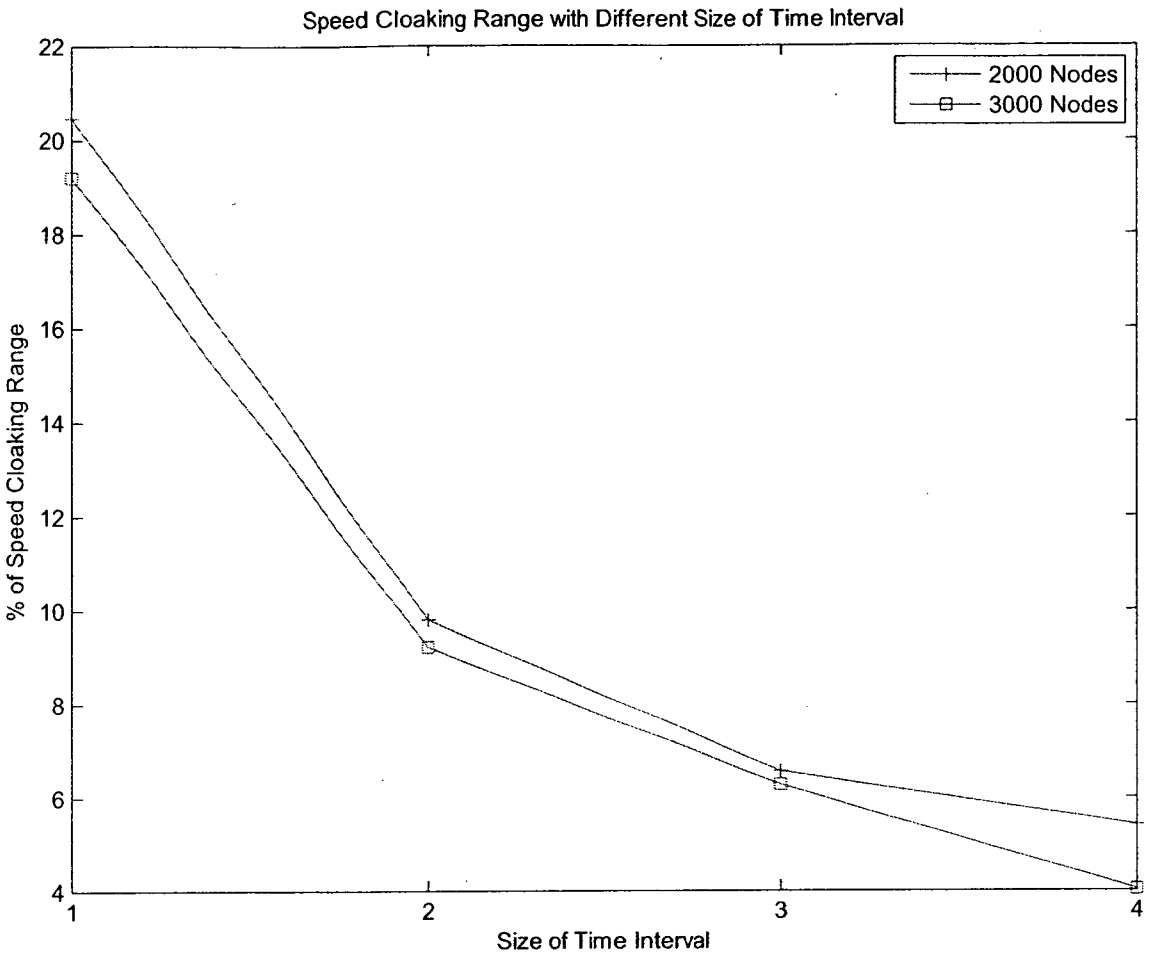

Figure 19: Speed Cloaking with Different Time Intervals

similar to what we obtain before. When the density of nodes increases, the cloaking range will become smaller leading to better service and a higher density of devices also causes cloaking success rate to increase.

We then integrate the simulatable auditing feature and conduct simulations to see how the cloaking changes under different constraints with the auditing enabled. As shown in Figure 27 and Figure 28, the results look different from the results shown in Figure 22 and Figure 23. The cloaking range decreases as constraint boxes become bigger partly due to that the cloaking success rate is also bigger. With the auditing enabled, when the constraint box is smaller, more devices that cannot be cloaked will be added to the denial group. The denial group is a group of devices that cannot be cloaked within its desired constraint

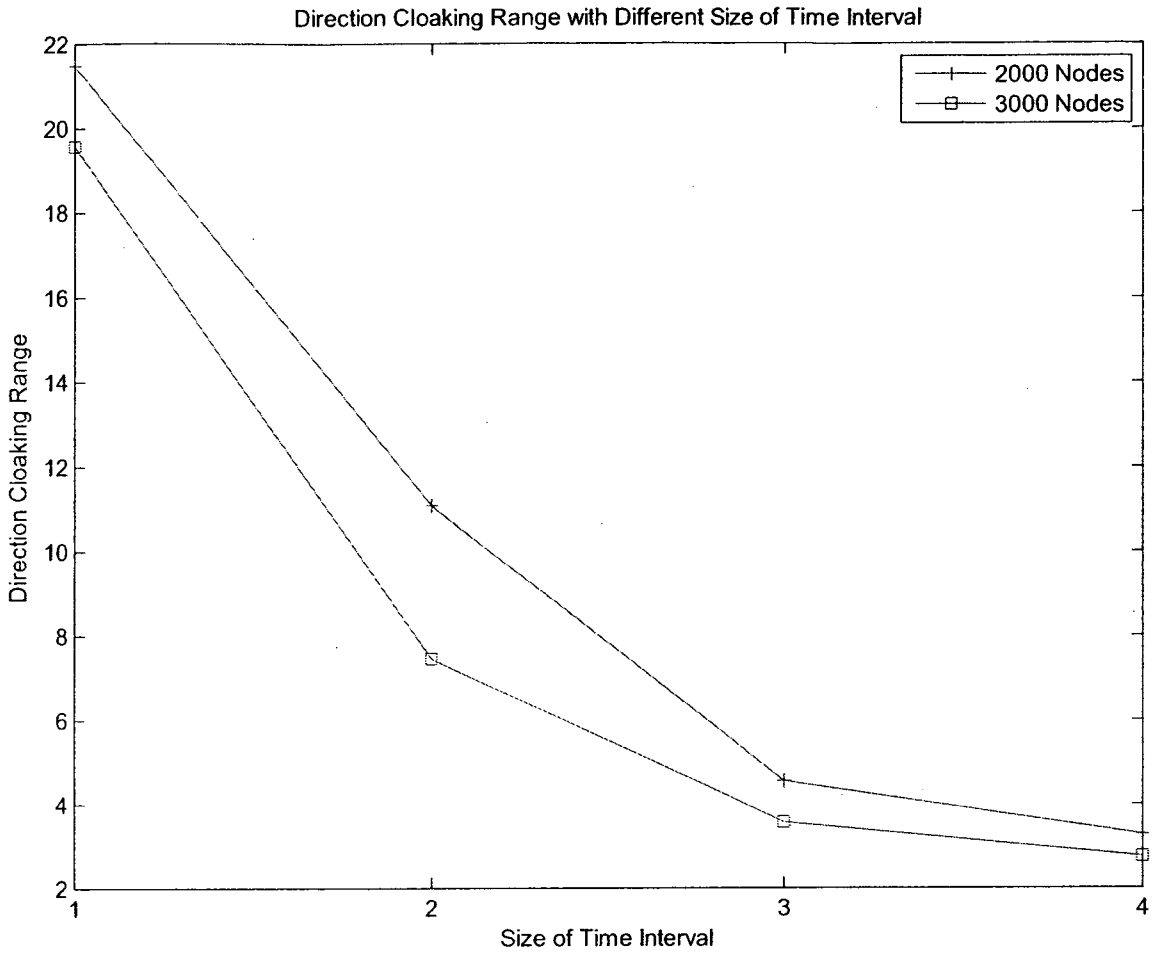

Figure 20: Direction Cloaking with Different Time Intervals

requirement, so the size of this special cloaking box is significantly bigger than normal cloaking boxes which leads to the overall increased cloaking result.

We then compare the cloaking method with and without customized constraints in Figure 29 and Figure 30. We use 3000 nodes, with constraint box of size 300 and anonymity requirement of 3. The results shown here indicate that by using the speed priority cloaking method, a device can have a better speed cloaking range.

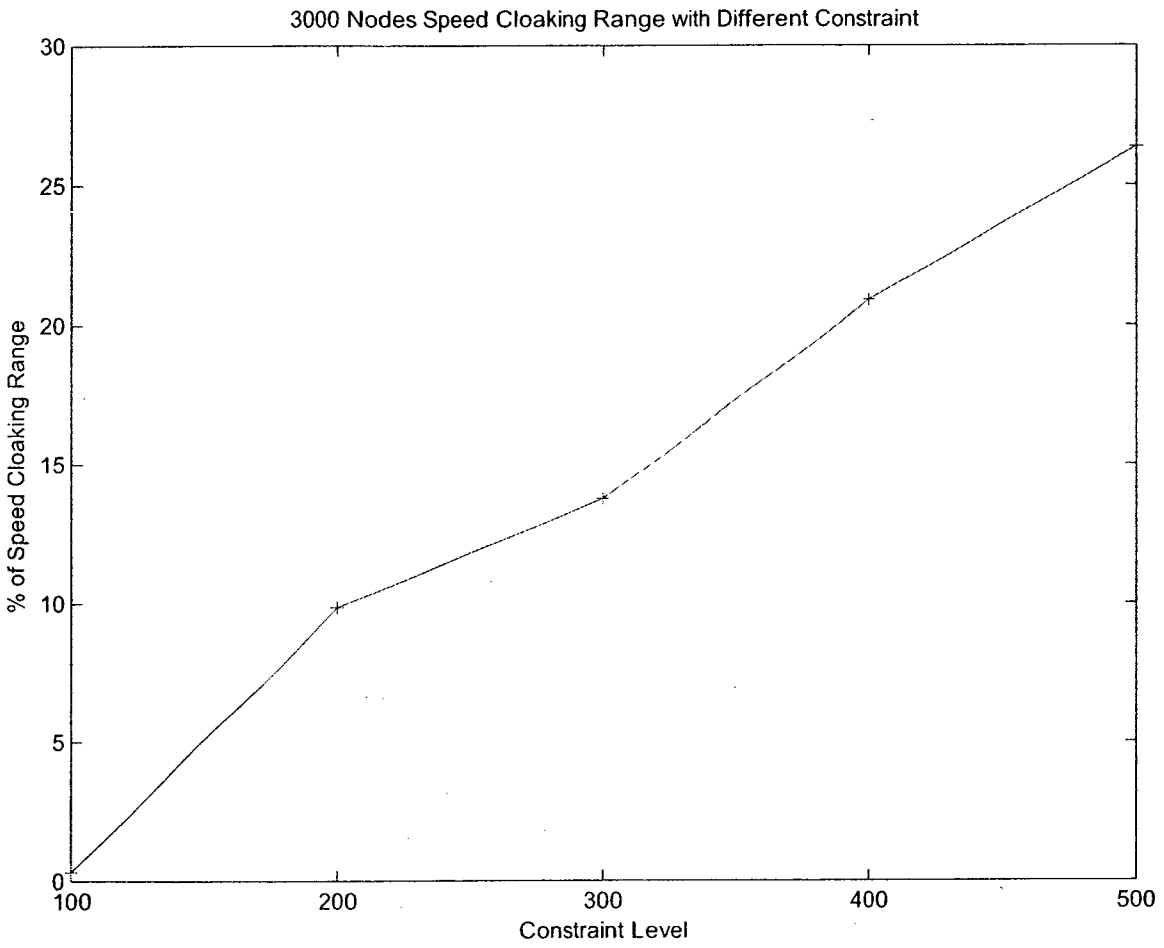

Figure 21: Speed Cloaking with Different Constraints

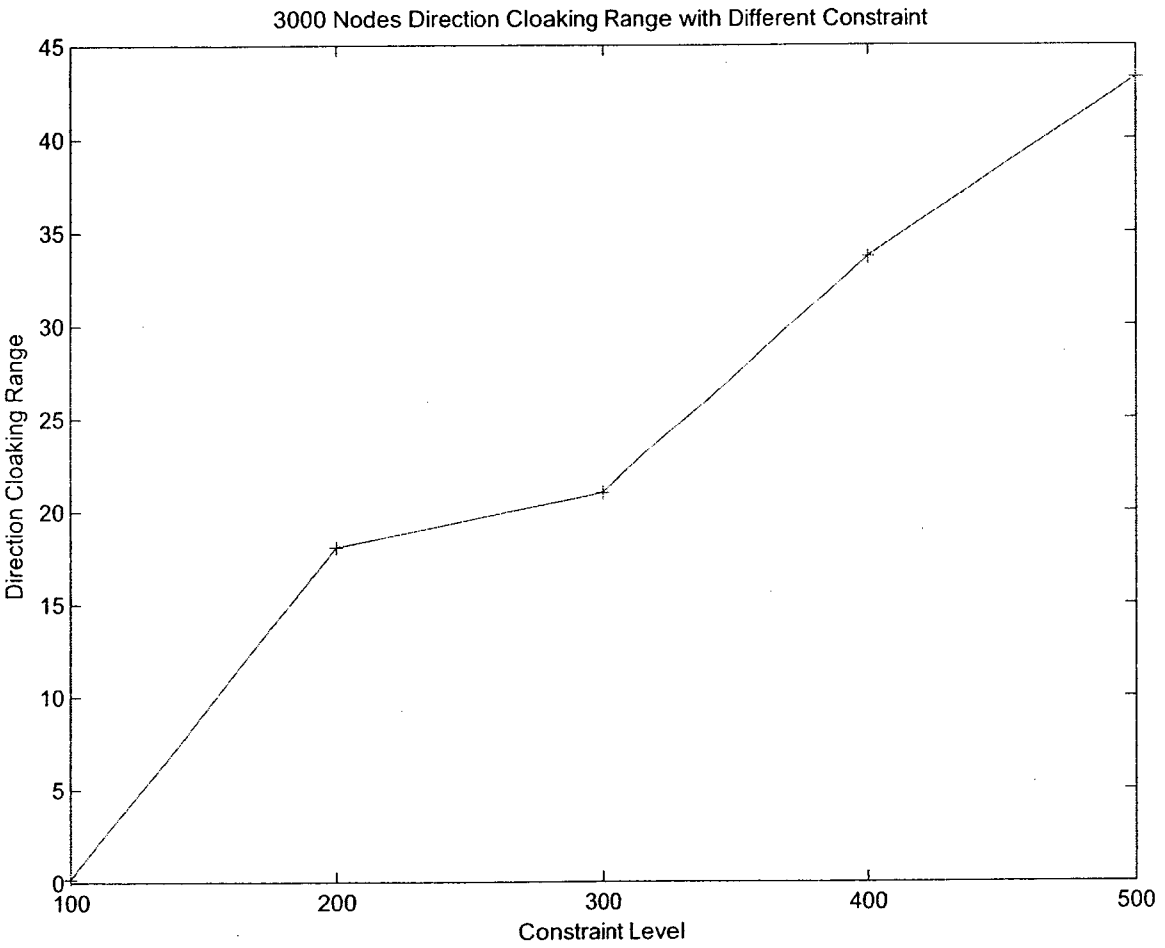

Figure 22: Direction Cloaking with Different Constraints

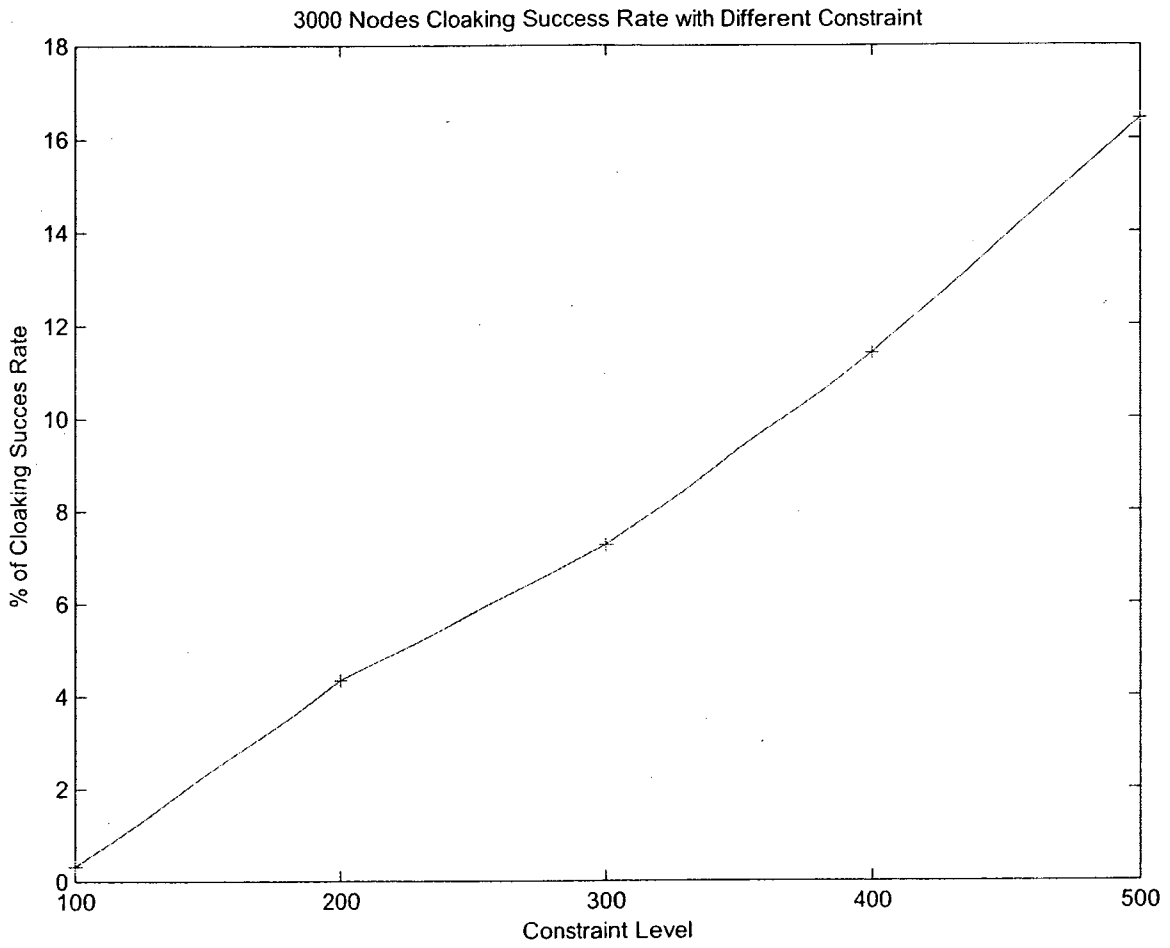

Figure 23: Cloaking Success Rate with Different Constraints

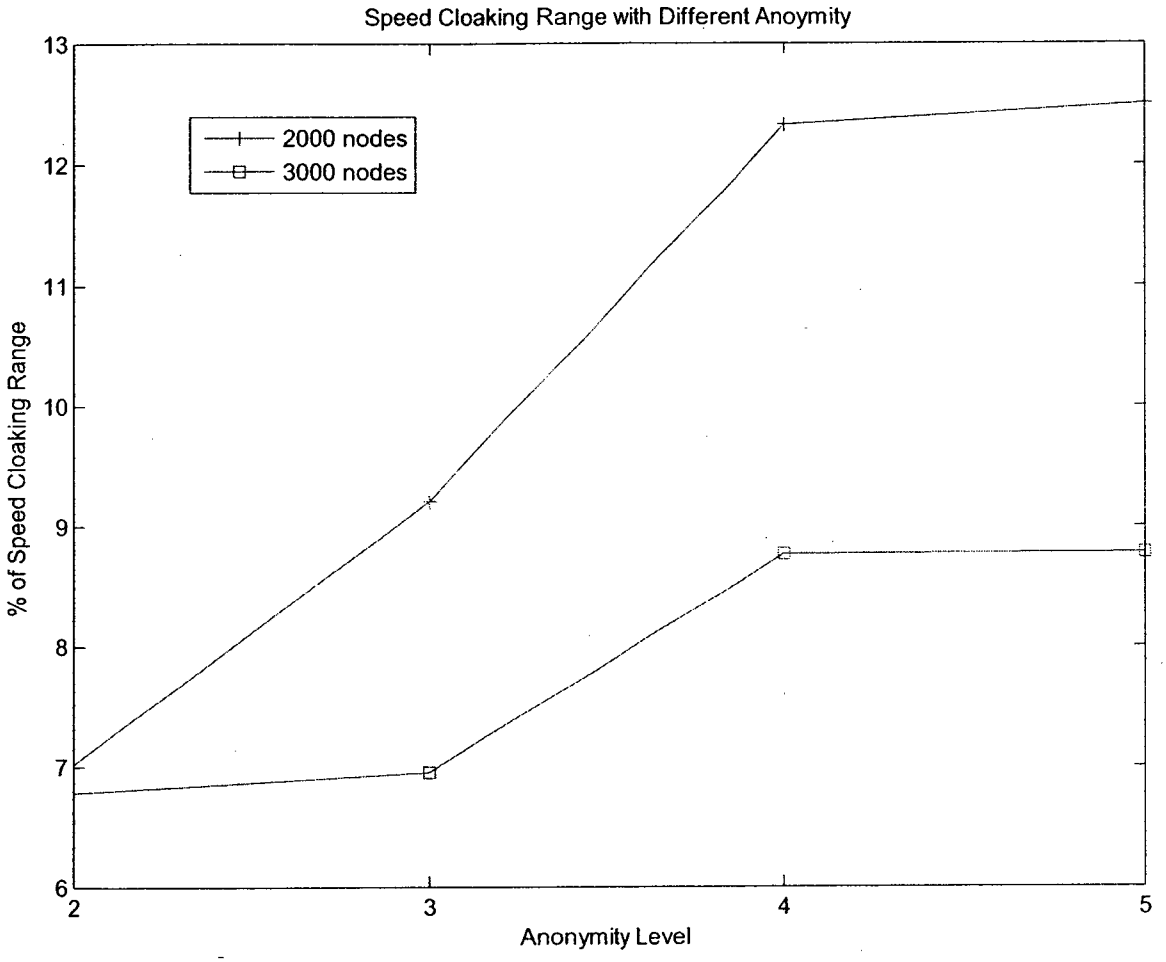

Figure 24: Speed Cloaking with Different Anonymity Requirements

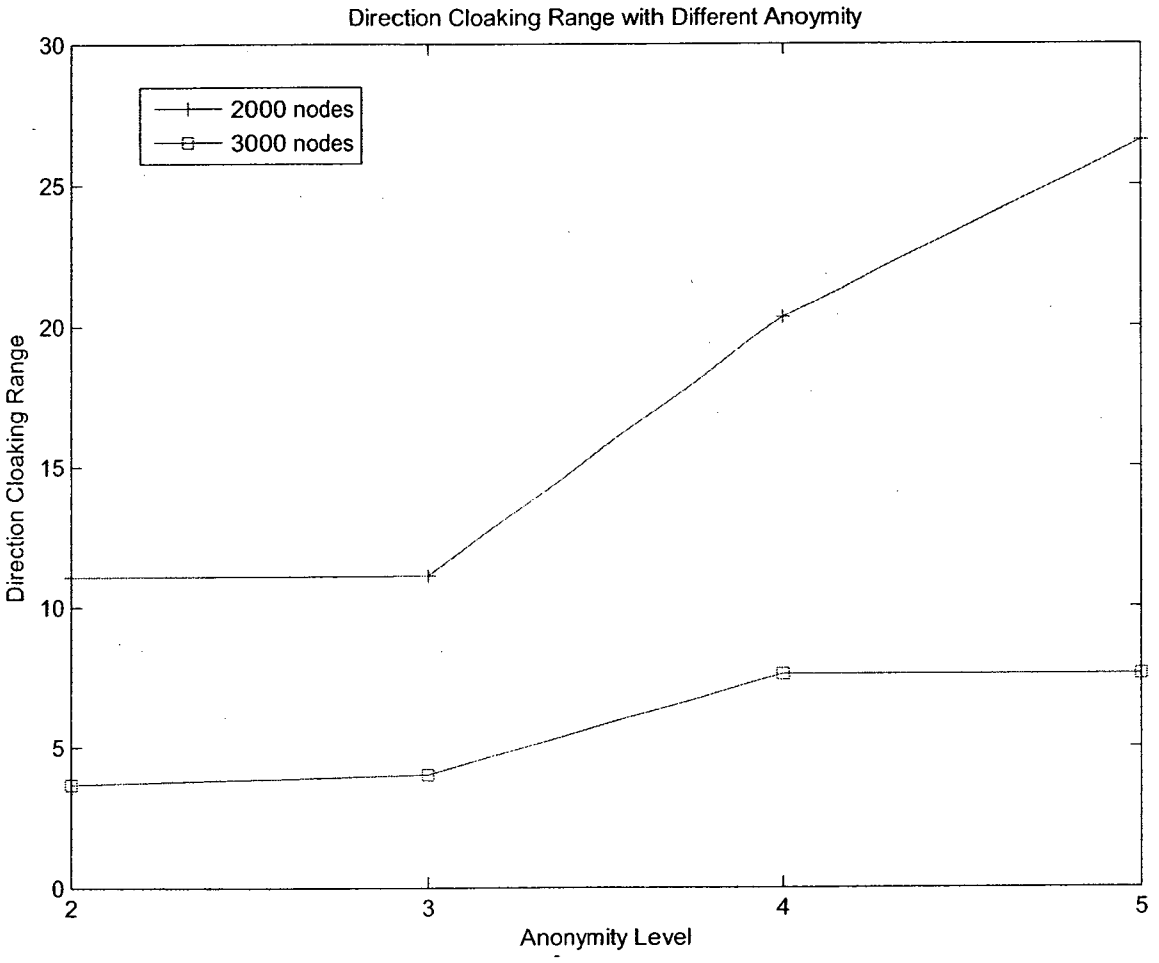

Figure 25: Direction Cloaking with Different Anonymity Requirements

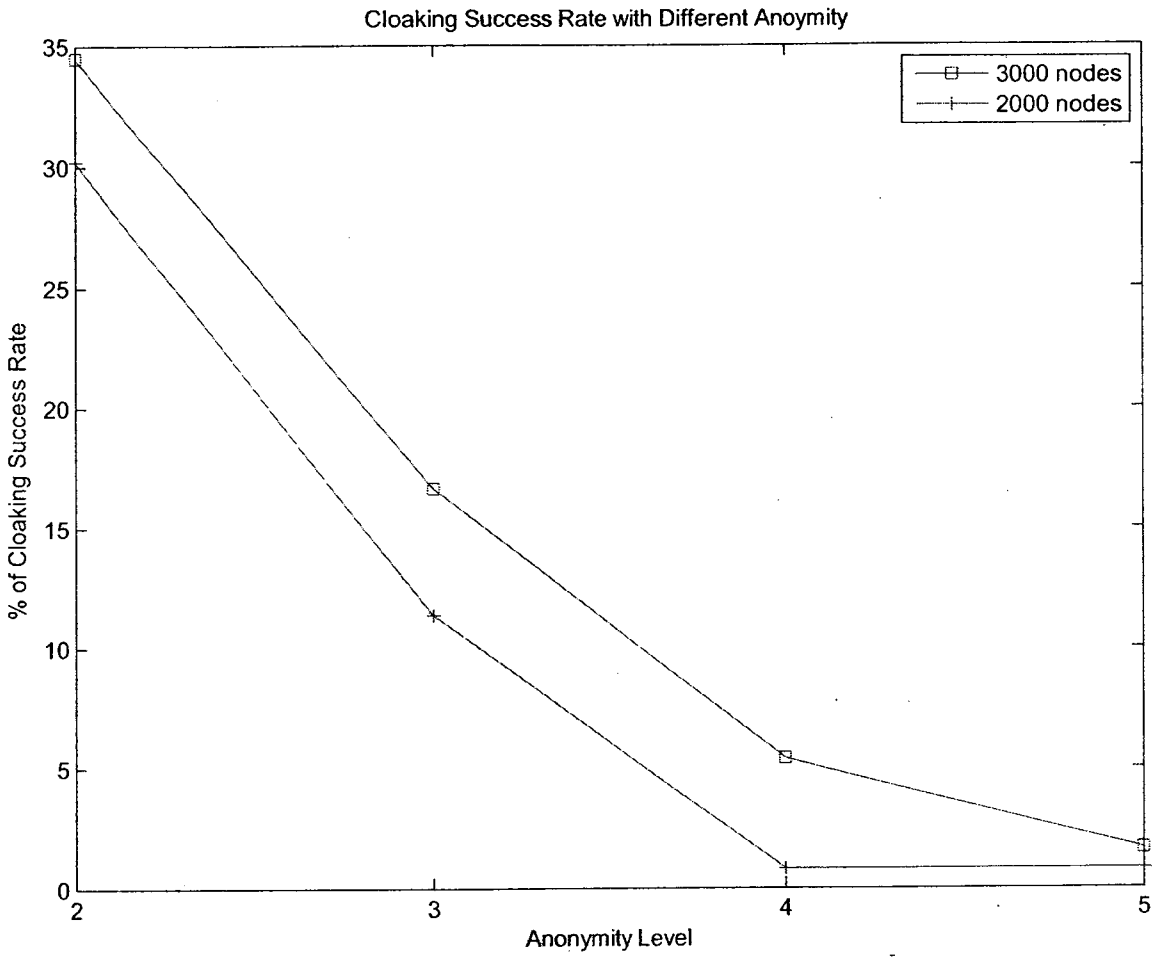

Figure 26: Success Rate With Different Anonymity Levels

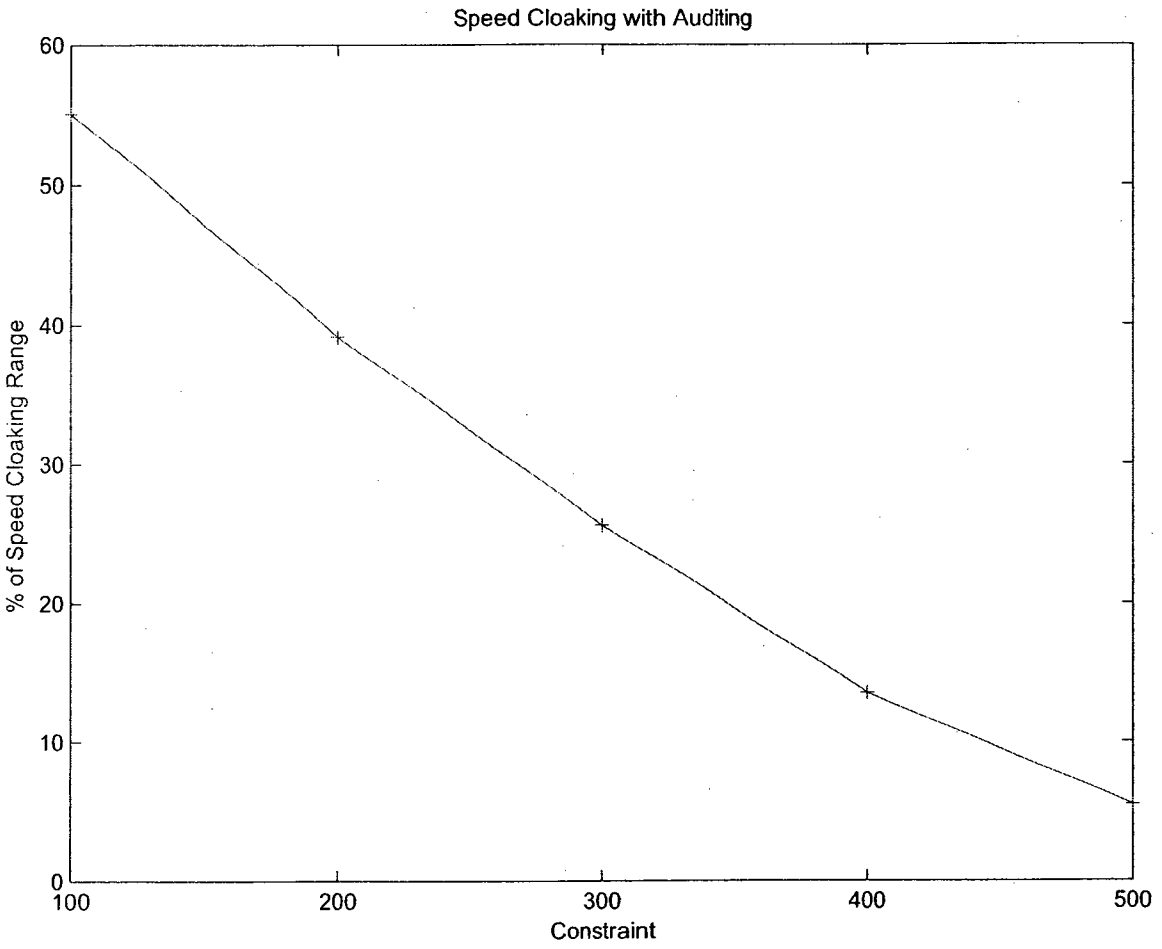

Figure 27: Speed Cloaking with Different Constraints

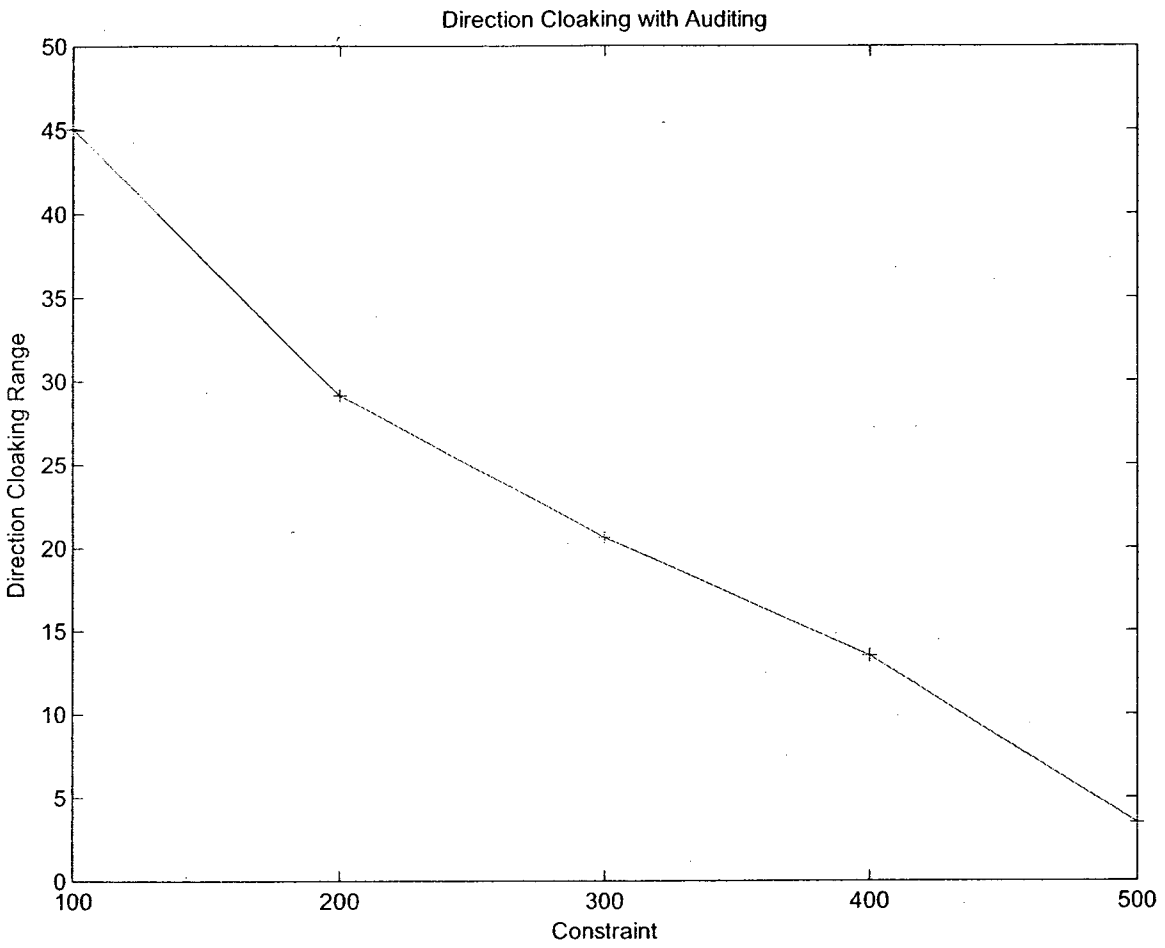

Figure 28: Direction Cloaking with Different Constraints

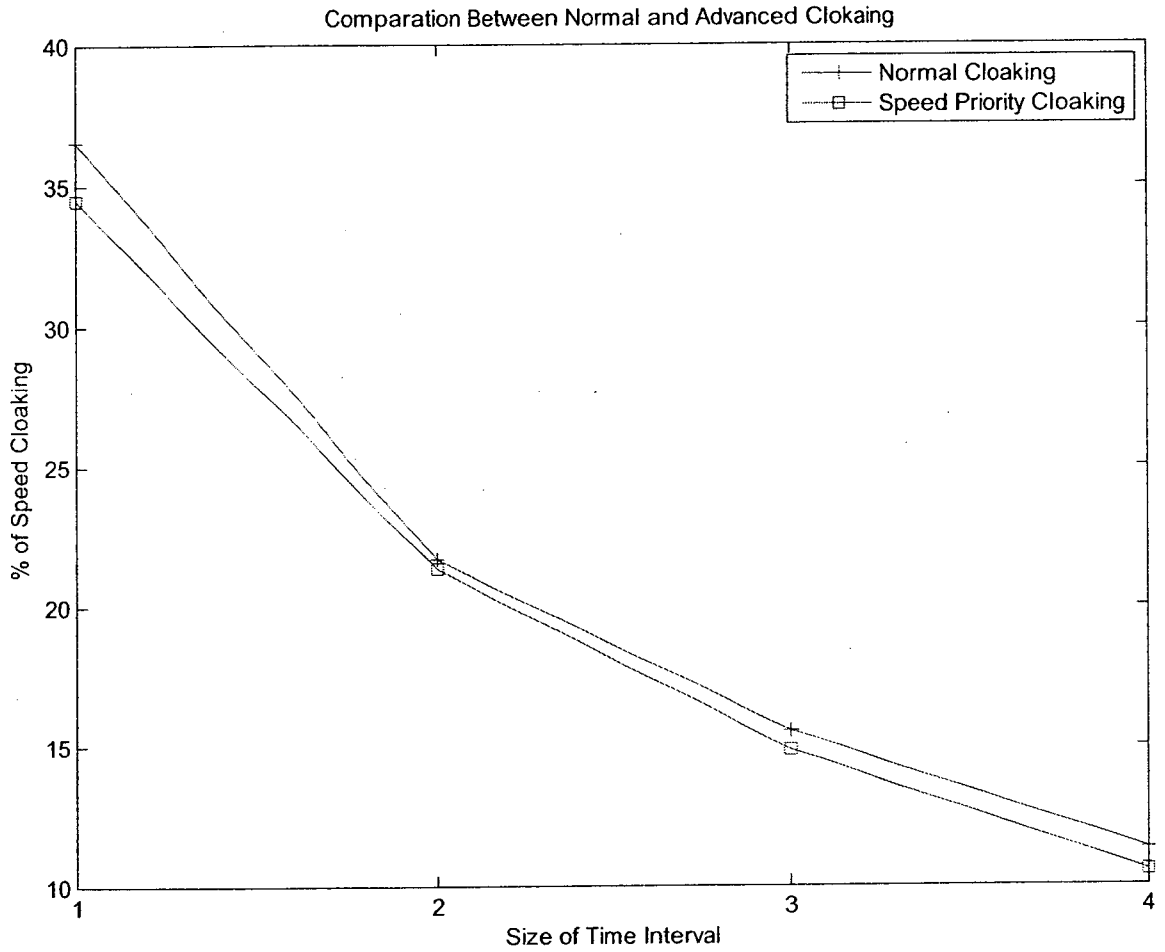

Figure 29: Comparison with Speed Cloaking

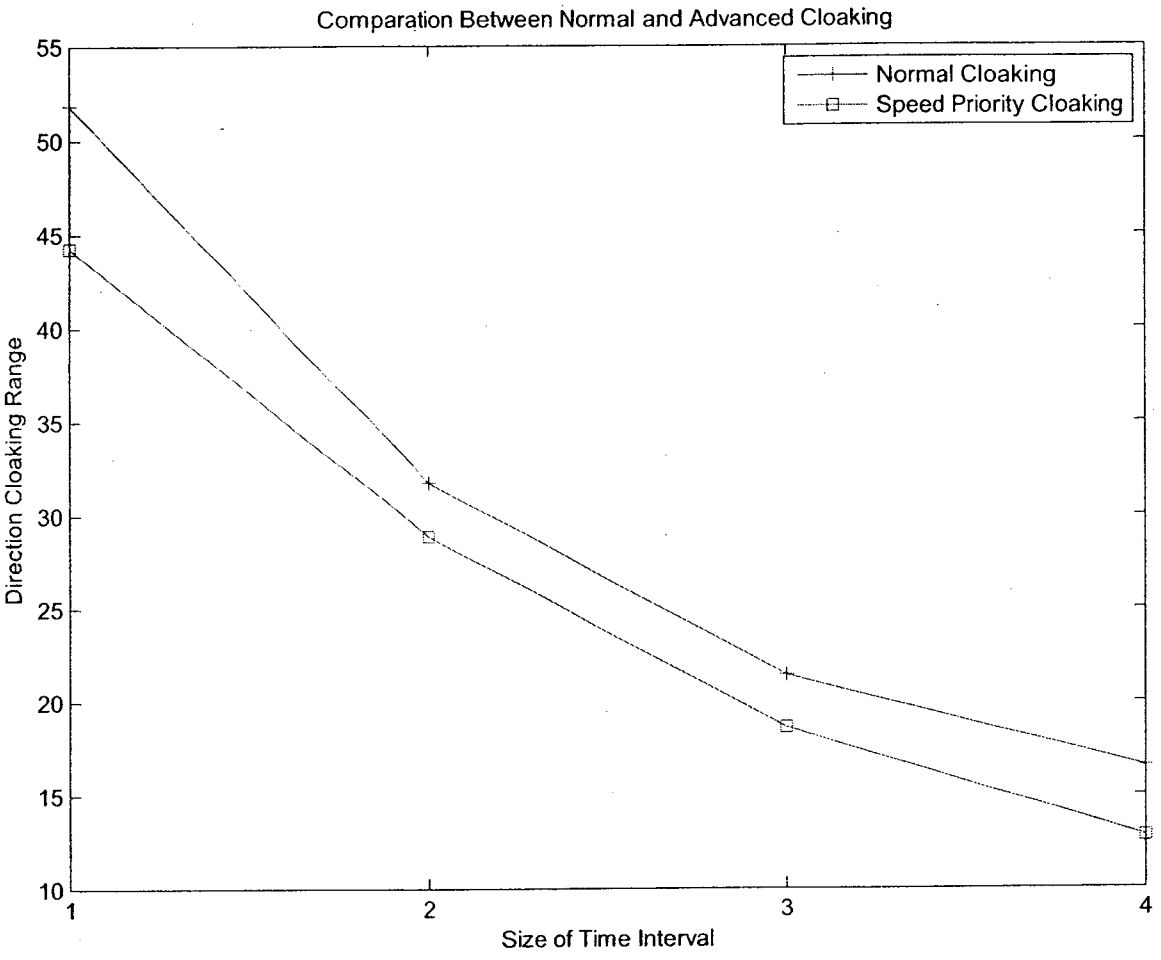

Figure 30: Comparison with Direction Cloaking

Chapter 7

Conclusion

We have addressed the privacy issue of LBS with continuous queries containing location data such as speed and direction. We showed that inference attacks using the extra information about speed and direction can defeat existing cloaking methods. We have discussed attacks under different assumptions about the availability of location data and in most cases, an adversary was able to improve his/her estimation of the actual location. We then showed that addressing such inference attacks requires cloaking both speed and direction. We have proposed a new method for cloaking speed and direction based on estimated future cloaking boxes. We then propose a customized speed and direction cloaking method to allow users to customize their preferences. We also propose to apply the simulatable auditing concept to location cloaking to guarantee that all mobile devices' privacy are always protected even when their constraints may not be satisfied by any cloaking. Finally, we have presented simulation results on proposed methods based on a real world city map. There are still few things that need to be improved in future work, such as further devising detailed methods to manage estimation errors in cloaking, the study of cloaking boxes of general geometric shapes, and the study of other types of cloaking constraints and preferences. We will also study the computational and storage cost of the proposed methods especially when the simulatable auditing is enabled.

Bibliography

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

[2] T. Brinkhoff. A framework for generating network-based moving objects. In *GeoInformatica*, volume 6(2), 2002.

[3] Alexander Brodsky, Lei Zhang, and Sushil Jajodia. *Answering Queries based on Imprecision and Uncertainty Trade-Offs in Numeric Databases*, volume volume 4721/2007. Springer Berlin and Heidelberg, 2007.

[4] B.Zheng, W.-C. Lee, and D.Lee. Search continuous nearest neighbors on the air. In *The 1st International Conference on Mobile and Ubiquitous Systems: Networking and Services*, pages 236–245, Boston,MA,USA, 2004.

[5] Francis Chin. Security problems on inference control for sum, max, and min queries. *J. ACM*, 33(3):451–464, 1986.

[6] Francis Chin and Gultekin Ozsoyoglu. Auditing for secure statistical databases. In *ACM 81: Proceedings of the ACM '81 conference*, pages 53–59, New York, NY, USA, 1981. ACM.

[7] Eun-Ae Cho, Chang-Joo Moon, Hyun-Soo Im, and Doo-Kwon Baik. An anonymous communication model for privacy-enhanced location based service using an echo

agent. In *ICUIMC '09: Proceedings of the 3rd International Conference on Ubiquitous Information Management and Communication*, pages 290–297, New York, NY, USA, 2009. ACM.

[8] Chi-Yin Chow and Mohamed F. Mokbel. Privacy in location-based services: a system architecture perspective. *SIGSPATIAL Special*, 1(2):23–27, 2009.

[9] Chi-Yin Chow, Mohamed F. Mokbel, and Xuan Liu. A peer-to-peer spatial cloaking algorithm for anonymous location-based services. In *ACM-GIS*, pages 171–178, New York, NY, USA, 2006. ACM.

[10] Alissa Cooper and John Morris. Binding privacy rules to location on the web. In *LOCWEB '09: Proceedings of the 2nd International Workshop on Location and the Web*, pages 1–4, New York, NY, USA, 2009. ACM.

[11] Claudia Diaz, Stefaan Seys, Joris Claessens, and Bart Preneel. Towards measuring anonymity, 2002.

[12] David Dobkin, Anita K. Jones, and Richard J. Lipton. Secure databases: protection against user influence. *ACM Trans. Database Syst.*, 4(1):97–106, 1979.

[13] Bugra Gedik and Ling Liu. Location privacy in mobile system: A personalized anonymization model. In *The 25th IEEE International Conference on Distributed Computing Systems*, pages 620–629, 2005.

[14] Bugra Gedik and Ling Liu. Protecting location privacy with personalized k-anonymity: Architecture and algorithms. *IEEE Transactions on Mobile Computing*, 7(1):1–18, 2008.

[15] Marius O. Gheorghita, Agusti Solanas, and Jordi Forné. Location privacy in chain-based protocols for location-based services. In *ICDT '08: Proceedings of the 2008*

The Third International Conference on Digital Telecommunications, pages 64–69, Washington, DC, USA, 2008. IEEE Computer Society.

[16] Gabriel Ghinita. Understanding the privacy-efficiency trade-off in location based queries. In *SPRINGL '08: Proceedings of the SIGSPATIAL ACM GIS 2008 International Workshop on Security and Privacy in GIS and LBS*, pages 1–5, New York, NY, USA, 2008. ACM.

[17] Gabriel Ghinita. Private queries and trajectory anonymization: a dual perspective on location privacy. *Trans. Data Privacy*, 2(1):3–19, 2009.

[18] Gabriel Ghinita, Panos Kalnis, Ali Khoshgozaran, Cyrus Shahabi, and Kian-Lee Tan. Private queries in location based services: anonymizers are not necessary. In *SIGMOD '08: Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 121–132, New York, NY, USA, 2008. ACM.

[19] Gabriel Ghinita, Panos Kalnis, and Spiros Skiadopoulos. Prive: Anonymous location-based queries in distributed mobile systems. In *WWW Track: Pervasive Web and Mobility*, pages 371–380, New York, NY, USA, 2007. ACM.

[20] Oded Goldreich. *Foundations of Cryptography*, volume Basic Tools. Cambridge University Press, 2001.

[21] Marco Gruteser and Dirk Grunwald. Anonymous usage of location-based service through spatial and temporal cloaking. In *MOBISYS*, pages 31 – 42, New York, NY, USA, 2003. ACM.

[22] Marco Gruteser and Baik Hoh. *Security in Pervasive Computing*. Springer Berlin / Heidelberg, 2005.

[23] Marco Gruteser and Xuan Liu. Protecting privacy in continuous location-tracking applications. *IEEE Security and Privacy*, 2:28–34, 2004.

[24] Urs Hengartner. Hiding location information from location-based services. In *MDM '07: Proceedings of the 2007 International Conference on Mobile Data Management*, pages 268–272, Washington, DC, USA, 2007. IEEE Computer Society.

[25] Baik Hoh and Marco Gruteser. Protecting location privacy through path confusion. In *SECURECOMM '05: Proceedings of the First International Conference on Security and Privacy for Emerging Areas in Communications Networks*, pages 194–205, Washington, DC, USA, 2005. IEEE Computer Society.

[26] Panos Kalnis, Gabriel Ghinita, Kyriakos Mouratidis, and Dimitris Papadias. Preserving anonymity in location based services. In *TRB*, 2006.

[27] John B. Kam and Jeffrey D. Ullman. A model of statistical database their security. *ACM Trans. Database Syst.*, 2(1):1–10, 1977.

[28] Krishnaram Kenthapadi, Nina Mishra, and Kobbi Nissim. Simulatable auditing. In *PODS '05: Proceedings of the twenty-fourth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 118–127, New York, NY, USA, 2005. ACM.

[29] Jon Kleinberg, Christos Papadimitriou, and Prabhakar Raghavan. Auditing boolean attributes. *J. Comput. Syst. Sci.*, 66(1):244–253, 2003.

[30] Lars Kulik. Privacy for real-time location-based services. *SIGSPATIAL Special*, 1(2):9–14, 2009.

[31] Po-Yi Li, Wen-Chih Peng, Tsung-Wei Wang, Wei-Shinn Ku, Jianliang Xu, and J. A. Hamilton Jr. A cloaking algorithm based on spatial networks for location privacy. *Sensor Networks, Ubiquitous, and Trustworthy Computing, International Conference on*, 0:90–97, 2008.

[32] Fuyu Liu, Kien A. Hua, and Ying Cai. Query l-diversity in location-based services. In *MDM '09: Proceedings of the 2009 Tenth International Conference on Mobile Data Management: Systems, Services and Middleware*, pages 436–442, Washington, DC, USA, 2009. IEEE Computer Society.

[33] Joseph T. Meyerowitz and Romit Roy Choudhury. Realtime location privacy via mobility prediction: creating confusion at crossroads. In *HotMobile '09: Proceedings of the 10th workshop on Mobile Computing Systems and Applications*, pages 1–6, New York, NY, USA, 2009. ACM.

[34] M.Gruteser and B.Hoh. On the anonymity of periodic location samples. In *Second International Conference on Security in Pervasive Computing*, 2005.

[35] Abedelaziz Mohaisen, Dowon Hong, and DaeHun Nyang. Privacy in location based services: Primitives toward the solution, 2009.

[36] Mohamed F. Mokbel, ChiYin Chow, and Walid G. Aref. The new casper: Query processing for location services without compromising privacy. In *THE 32nd International Conference on Very Large Data Bases*, pages 763–774, 2006.

[37] Nishkam Ravi, Marco Gruteser, and Liviu Iftode. Non-inference: An information flow control model for location-based services. *Mobile and Ubiquitous Systems, Annual International Conference on*, 0:1–10, 2006.

[38] Steven P. Reiss. Security in databases: A combinatorial study. *J. ACM*, 26(1):45–57, 1979.

[39] Andrei Serjantov, Andrei Serjantov, and George Danezis. Towards an information theoretic metric for anonymity. pages 41–53. Springer-Verlag, 2002.

[40] Heechang Shin, Vijayalakshmi Atluri, and Jaideep Vaidya. A profile anonymization model for privacy in a personalized location based service environment. In *MDM '08:*

Proceedings of the The Ninth International Conference on Mobile Data Management, pages 73–80, Washington, DC, USA, 2008. IEEE Computer Society.

[41] Agusti Solanas and Antoni Martínez-Ballesté. A ttp-free protocol for location privacy in location-based services. *Comput. Commun.*, 31(6):1181–1191, 2008.

[42] Yan Sun, Thomas F. La Porta, and Parviz Kermani. A flexible privacy-enhanced location-based services system framework and practice. *IEEE Transactions on Mobile Computing*, 8(3):304–321, 2009.

[43] Jungho Um, Hyeongil Kim, Youngho Choi, and Jaewoo Chang. A new grid-based cloaking algorithm for privacy protection in location-based services. In *HPCC '09: Proceedings of the 2009 11th IEEE International Conference on High Performance Computing and Communications*, pages 362–368, Washington, DC, USA, 2009. IEEE Computer Society.

[44] Zhen Xiao, Jianliang Xu, and Xiaofeng Meng. p-sensitivity: A semantic privacy-protection model for location-based services. In *MDMW '08: Proceedings of the 2008 Ninth International Conference on Mobile Data Management Workshops*, pages 47–54, Washington, DC, USA, 2008. IEEE Computer Society.

[45] Fei Xu, Jingsha He, Xu Wu, and Jing Xu. A method for privacy protection in location based services. *Computer and Information Technology, International Conference on*, 2:351–355, 2009.

[46] Toby Xu and Ying Cai. Location anonymity in continuous lcation-based services. In *The 15th international Symposium on Advances in Geographic Information Systems*, 2007.

[47] Toby Xu and Ying Cai. Exploring historical location data for anonymity preservation in location-based services. In *The 27th Conference on Computer Communications. IEEE*, 2008.

[48] Man Lung Yiu, Christian S. Jensen, Xuegang Huang, and Hua Lu. Spacetwist: Managing the trade-offs among location privacy, query performance, and query accuracy in mobile services. In *ICDE '08: Proceedings of the 2008 IEEE 24th International Conference on Data Engineering*, pages 366–375, Washington, DC, USA, 2008. IEEE Computer Society.

[49] Y.Tao, D. Papadias, and Q.Shen. Continuous nearest neighbor search. In *Proceedings of International Conference on Very Large Data Bases(VLDB'02)*, pages 20–23, HongKong,China, 2002.

[50] Chengyang Zhang and Yan Huang. Cloaking locations for anonymous location based services: a hybrid approach. *Geoinformatica*, 13(2):159–182, 2009.

[51] Ge Zhong, Ian Goldberg, and Urs Hengartner. Louis, lester and pierre: Three protocols for location privacy. In *Privacy Enhancing Technologies*, pages 62–76, 2007.

[52] Z.Song and N. Roussopoulos. K-nearest neighbor search for moving query point. In *Proceedings of the 7th International Symposium on Advances in Spatial and Temporal Databases(SSTD '01)*, pages 79–96, London,UK, 2001.