2012 IEEE 18th International Conference on Parallel and Distributed Systems

Complete Bipartite Anonymity: Confusing Anonymous Mobility Traces for Location Privacy

Kai Dong*, Tao Gu†, Xianping Tao*, Jian Lu*
*State Key Laboratory for Novel Software Technology, Nanjing University
Email: kaidong@smail.nju.edu.cn, txp@nju.edu.cn, lj@nju.edu.cn.
†University of Southern Denmark
Email: gu@imada.sdu.dk

***Abstract*—Using mobile devices, people can easily obtain their location information, and access a wide range of location based services (LBSs). Many existing LBSs rely on accurate, continuous, and real-time streams of location information to provide quality of service guarantees. In this case, even if an user accesses LBSs anonymously, the identity of the user can still be revealed by analyzing the mobility trace. To protect user privacy, existing work sacrifice the quality of LBSs by degrading spatial and temporal accuracy. To achieve a better tradeoff between user privacy and the quality of service, we present a novel approach, Complete Bipartite Anonymity (CBA), to confuse the paths of nearby users by connecting different users' real traces with fake ones. CBA protects user privacy as users become indistinguishable after their paths are confused; the quality of service of LBSs is also guaranteed since users are able to report their accurate locations. We evaluate CBA by comparing the system and privacy performance with existing techniques such as Path Confusion or Query Obfuscation using a real-world data set, the results show that our scheme increases the chance for a user joining an anonymity group by 10 times in low user density areas, and reduces the resources consumed by about 90% for achieving the same anonymity degree.**

***Keywords*-LBS, Location Privacy, Path Confusion, Query Obfuscation, Complete Bipartite Anonymity.**

I. Introduction

With the advances of global positioning system (GPS) and mobile devices, people can easily obtain their location information, and access a wide range of location based services (LBSs). In LBSs, users have to reveal their locations to location based service providers (LSPs). This poses a significant privacy risk since LSPs may disclose users' location information to others.

Anonymous communication techniques can be used to increase users' location privacy. In these techniques, user ID is typically omitted and the network address problem is addressed by mechanisms such as onion routing [1], to ensure sender anonymity. However, revealing anonymous location information poses new problems [2]. The anonymous locations can be classified by their temporal and spatial relationships, to form traces of different users, much like a trial of the breadcrumbs left by Hansel and Gretel in the popular tale [3]. An adversary may mount an identification attack to identify and track the subject if the breadcrumb trial contains some "identifying locations[1]".

Two different approaches have been proposed to prevent identification attacks. Spatial and temporal cloaking [2] uses location blurring in which an accurate location is blurred into an area and all the requests from this area within a certain period of time are managed together to achieve anonymity. Other work use the same concept in different settings such as [4], [5], [6]. Another approach [7] leverages on the concept of Mix Zones to achieve user anonymity. A Mix Zone is generated if there are enough users located in the same place at the same time. Since no location information is reported when users are in a Mix Zone, their traces are "mixed". As such, the LSP cannot distinguish a user from others who are in the zone at the same time, and also cannot link users entering the zone with those coming out of it. The concept of Mix Zones has been widely adopted in [8], [9], [10].

However, these techniques have limitations since they sacrifice the accuracy of user information. Suppose a LBS which provides users the traffic information near their locations, it relies on an accurate, continuous, and real-time stream of location information to provide Quality of Service (QoS) guarantees. The aforementioned techniques do not guarantee the desired QoS – the cloaking based techniques degrade spatial accuracy and increase delay in reporting users' locations; the Mix Zone based techniques temporarily prevent users from reporting their locations in the zone area, resulting in the loss of their accurate locations in the zone.

In this paper, we present a novel approach – Complete Bipartite Anonymity (CBA) – to achieve a better tradeoff between user privacy and the quality of service. The intuition is simple. We group nearby users into a special region, named CBA zone, in the way that the users enter or leave the region at the same time. Our idea is to confuse the paths of the users in this region by connecting their query traces while allowing location reporting for each user as usual. In a CBA zone, a user not only reports his real trace, but also

[1]Identifying locations are locations where the subject can be identified [3]. For example, the LBS is invoked at the time a user is in the garage, the location coordinate can be mapped to the address of the owner of the residence [2]. In this case, the garage is an identifying location.

1521-9097/12 $26.00 © 2012 IEEE
DOI 10.1109/ICPADS.2012.37

generate fake query traces connecting to the traces of all other users. Each CBA zone can be viewed as a complete bipartite graph where each entry or exit points is a vertex and there exists a query trace (i.e., edge) for each pair of <*entry*, *exit*>. We say a group of nearby users satisfies complete bipartite anonymity if and only if a CBA zone is a complete bipartite graph. For a user group satisfying CBA, user privacy is protected by obfuscating the path of each user with that of all other users in a CBA zone; the QoS of LBSs is achieved since every user is able to query LBSs with his accurate, continuous and real-time location information.

Although the idea of CBA works in principle, developing a realistic scheme is a non-trivial task. The first question is that how nearby users find each other to create a CBA zone as far as privacy is concerned. The LSP may not be trustworthy and the communication channel to the LSP may not be secure. As a result, generating a CBA zone based on information provided by LSPs may violate user privacy. In CBA, we propose a collaborative path confusion mechanism to enable nearby users work together to generate a CBA zone without involving any trusted entities.

Secondly, how to ensure the fake traces resemble the real ones is a difficult task. Recent work have demonstrated that fake traces can be distinguished from realistic user move patterns, e.g., Peddinti et al. [11] present a classification attack that can identify up to 93.67% of real user trips from a data set with 5 times fake user trips. To address this problem, we propose a cloaked obfuscation method which makes the real user move patterns indistinguishable from fake ones.

The rest of the paper is organized as follows. We first discuss the related work in Section 2. We then describe the idea of CBA in Section 3, and the detailed scheme of CBA in Section 4, 5, and 6, respectively. Section 7 presents the experimental evaluation on system performance and privacy performance. Section 8 concludes the paper.

II. Related Work

Much work have been done leveraging on k-anonymity – an aggregation of k users satisfies k-anonymity if every user is indistinguishable. The probability of identifying a user from this aggregation is $1/k$. To achieve k-anonymity, a suppression or generalization function can be used [12]. However, simply applying such function may reduce QoS. Gruteser et al. [2] used k-anonymity to increase location privacy, and proposed spatial and temporal cloaking. In this technique, all the requests (at least k different users) from an area within a certain period of time are managed together as an anonymity set to achieve k-anonymity. To aggregate and transmit queries, a trusted third party (TTP) is required, which is named Anonymizer. The idea of anonymizing has been widely adopted, e.g., in [4] [5] [6] [13] [14]. Other approaches have been proposed to address the single point of failure problem raises from the TTP by providing a peer-to-peer (P2P) method to achieve distributed spatial cloaking [15] [16].

Table I: Notation used in this paper

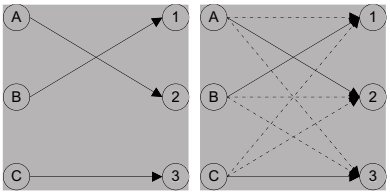
Term	Definition
$\mathcal{U}$	User group
$\mathcal{Z}$	Anonymity zone
$\mathcal{G}$	Trace graph
$\mathcal{L}$	Location
$\mathcal{T}_R$	Real trace
$\mathcal{T}_O$	Obfuscating trace
$\mathcal{T}_P$	Predicted trace
$\mathcal{S}_c$	Sequence of coordinates composing a trace
$\mathcal{S}_t$	Sequence of time stamps composing a trace

Several anonymizing algorithms have been proposed with the motivation that path suppression in high density areas increases the chance for confusing or mixing several different traces. Beresford et al. [7] first introduced a-priori defined Mix Zones to achieve user anonymity. A Mix Zone exists if there are enough users located in the same place at the same time. Users within the same Mix Zone serve as an anonymity set. Palanisamy et al. [10] proposed MobiMix, which breaks the continuity of location exposure by using Mix Zones so that no applications can trace user movement. Path Confusion [8] extends Mix Zones by perturbing location samples. The key idea underlying the perturbation algorithm is to cross paths in areas where at least two users meet. This technique increases the chance of confusion in high-density areas, but it also cannot guarantee privacy in low-density areas. Path Cloaking [9] has been proposed to improve Path Confusion. It resolves the same-place same-time problem, but one incorporates a delay in the anonymization, and the delay can be very long in low-density areas. This delay introduced in every crossing sacrifices QoS and arises new problems in realtime situations.

A different approach to increase location privacy is Query Obfuscation, which involves reporting $k-1$ fake locations or dummies along with the real path of a user. Kido et al. [17] proposed the concept of dummies to achieve k-anonymity when user density is low. They focused on reducing computation cost, and used the random walk model to generate dummy path. Other work leverage on data mining methods such as a probabilistic model [18] and a statistical clustering technique [19], for dummy path generation. Query Obfuscation has two limitations. The first limitation is wasting resources – users have to query $k-1$ extra time to achieve k-anonymity, and the LSPs have to process and response to all these fake queries. Although the overhead may be acceptable in the realm of bits, the cost will be too high for real-world services [7]. Second, how to ensure that dummies will not be distinguished from realistic user move patterns remains a challenging issue to solve.

III. Complete Bipartite Anonymity

In this section, we first give definition of CBA, we then use an example to show how CBA achieves the balance between privacy and QoS.



(a) (b)

Figure 1: Complete Bipartite Anonymity. In a CBA zone, (a) presents the real trace of user A, B and C, respectively, represented by solid line; (b) presents a complete bipartite graph, where obfuscating traces are generated, represented by dotted line.

A. Definitions

Based on the breadcrumb effect, a user's real queries can be classified and linked to form a real trace. A trace $\mathcal{T}$ is composed of a sequence of coordinates, $\mathcal{S}_c$ and a sequence of corresponding time stamps $\mathcal{S}_t$. To confuse this real trace, we use the query obfuscation technique to generate several fake queries to form a obfuscating trace.

For a group of users $\mathcal{U}$, each user has a different trace. The distance between any two traces can be measured by two thresholds: the spatial distance d_S and the temporal distance d_T. Based on these thresholds, we give the following definitions.

Definition 3.1 (zone): A zone $\mathcal{Z}$ is a three dimensional area – two represent the coordinates and the other represents the time stamp. It covers several sub-traces of different users which are close enough.

Definition 3.2 (trace graph): In a zone $\mathcal{Z}$, take the first and last coordinate and time stamp of each trace as vertices, take all the real traces and the obfuscating traces as edges, we get the trace graph $\mathcal{G}_\mathcal{Z}$ of $\mathcal{Z}$.

Definition 3.3 (complete bipartite anonymity): If the trace graph of a zone is a complete bipartite graph, we name this zone a complete bipartite anonymity (CBA) zone. We name the user group in a CBA zone satisfies complete bipartite anonymity a CBA group.

As illustrated in Fig. 1, in a CBA zone, a group of users satisfying CBA generates obfuscating traces to confuse real traces of each other, ensuring user privacy. Meanwhile, each user is still able to query LBSs with his accurate, continuous and real-time location information, achieving the QoS of the LBSs.

IV. Collaborative Path Confusion

In this section, we describe how CBA works in principle. We propose a collaborative path confusion (CPC) mechanism for nearby users $\mathcal{U}$ to work collaboratively to create a CBA zone. In CBA, we do not assume the LSP is a trustworthy entity. In addition, the CBA service provider (CBA-SP) is untrustworthy – it can be any third party which allows users to store information, or even the LSP itself.

In the CPC mechanism, when a user queries LBSs with his accurate location, he should also send his location[2] to the CBA-SP. The CBA-SP stores the user's pseudonym and location, and can provide information to others nearby. Through the CBA-SP, these users jointly establish a shared symmetric key by using the Diffie-Hellman Key Exchange, and exchange their exit points in cypher text which are predicted by the local prediction engine. Connecting one's entry point to other users' exit points, one generates several obfuscating traces. If each of these users queries with both the real traces and the obfuscating traces, they eventually form a group which satisfies CBA.

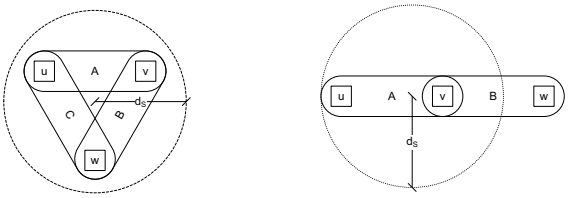
A. CPC Methods

We now describe the methods to implement the CPC mechanism. We start by a two-user case.

1) $Query : \mathcal{L}_f \rightarrow \mathcal{R}, Boolean.$
   This method allows a user $u \in \mathcal{U}$ to query LBSs along his path. This method takes location $\mathcal{L}_{fi}$ as input, and outputs result $\mathcal{R}_i$ satisfying $\mathcal{F}_{LBS}(\mathcal{L}_{fi}) = \mathcal{R}_i$ and a boolean value indicates whether there are other users nearby.
2) $InfoEx : u.String \rightarrow v.String$
   This method allows a user u to exchange information with another user v, without any other entities knowing this information. This method takes string $u.s_i$ generated by u as input, and outputs a corresponding string $v.s_j$ generated by v. They use the Diffie-Hellman Key Exchange to establish a shared symmetric key, and each generates a new pseudonym, u' for u and v' for v, respectively. Then u and v exchange their new pseudonyms in cypher-text.
3) $Setup : \mathcal{L}_f \rightarrow \mathcal{T}_O$
   This method allows two users, u and v to create a CBA zone. For each user, this method takes both users' locations $u.\mathcal{L}_{fi}$ and $v.\mathcal{L}_{fj}$ as input, and outputs an obfuscating trace.

Now we describe how the CPC mechanism works using the aforementioned methods. A user u, he takes a loop to call the *Query* method to access to LBSs along his path. Based on his location, the CBA-SP notifies him when there is another user (i.e., v) nearby. In this case, u calls the *InfoEx* method to exchange with v each other's newly generated pseudonym and location information. Then u calls the *Setup* method to exchange his pseudonym and to generate an obfuscating trace. Meanwhile, v also calls the *Setup* method, and a 2-anonymity CBA zone is created for both u and v. In this CBA zone, each user continues the loop to call the *Query* method, and queries with two locations, one from the real trace and the other from the obfuscating trace.

[2]To increase privacy, the location sent to the CBA-SP can be generalized to an area with existing spatial cloaking techniques such as [2].



(a) (b)

Figure 2: CBA zones with multiple users: In (a), users u, v, w are nearby users, they can generate a 3-anonymity CBA zone which is composed of three 2-anonymity CBA zones. In (b), since users u, w are not close enough, they can not generate a 3-anonymity CBA zone.

B. CPC Mechanism for Multiple Users

In the situation where there are multiple users in a CBA zone, i.e., k users nearby, we now illustrate that how they follow the CPC mechanism to create a k-anonymity CBA zone. Figure 2a depicts an example of 3-anonymity CBA zone. For users u, v and w, each pair combination generates a 2-anonymity CBA zone, i.e., u and v generate zone A, v and w generate zone B, u and w generate zone C. Combining these three zones, we obtain a 3-anonymity CBA zone, represented as the large circle. In contrast, if u and w are not close enough, they will not be able to generate a 3-anonymity CBA zone, as illustrated in Fig. 2b.

V. CBA Zone Generation

CBA zones are dynamically generated when the *Setup* method is invoked. From Definition 3.3, we know a CBA zone is composed of a spatial-temporal zone $\mathcal{Z}$, and a trace graph $\mathcal{G}_{\mathcal{Z}}$ which is a complete bipartite graph. To generate a CBA zone, users should first decide the boundary of $\mathcal{Z}$, and then generate all the traces composing $\mathcal{G}_{\mathcal{Z}}$.

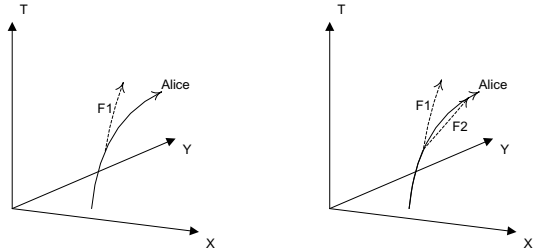
We now describe how a 2-anonymity CBA zone is generated. The boundary of a CBA zone covers all the entries and exits. Suppose users u and v are ready to generate a CBA zone. Their current locations are basically the two entry points. The boundary of zone $\mathcal{Z}$ is calculated to meet the following criteria. First, $\mathcal{Z}$ is restricted to the reasonable speed $\delta_{\mathscr{S}}$ and the reasonable direction δ_θ for both users, i.e., the exit boundary is reachable to both of them. Second, the size of this zone is restricted to the computational power.

After the boundary of the CBA zone is decided, u and v each generates an obfuscating trace $\mathcal{T}_O$ by using a trace generation method.

$$\mathcal{F}_{TraceGen} : \mathcal{L}, t, f \rightarrow \{\mathcal{T}\}$$

This method takes the start location $\mathcal{L}_s$, the end location $\mathcal{L}_e$, the corresponding time stamps t_s, t_e, and a querying frequency f as inputs, and outputs a trace $\mathcal{T} = (\mathcal{S}_c, \mathcal{S}_t)$.

We use the Microsoft Multimap API [20] to implement the trace generation algorithm. The API takes the source and the destination pair $< \mathcal{L}_s, \mathcal{L}_e >$ as inputs, and generates the shortest path between them. To generate the obfuscating trace $\mathcal{T}_O$, a user u sets the value of $\mathcal{L}_s$ to his current location $u.\mathcal{L}_{fi}$ (u's entry point), and set the value of $\mathcal{L}_e$ to the other user's future location (v's exit point). This location is calculated by v using the Microsoft Multimap navigation based on his target destination, his driving direction, and his current location.



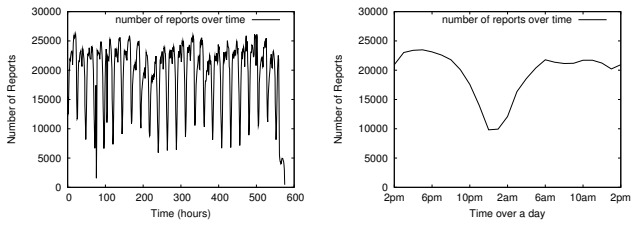
(a) (b)

Figure 3: Cloaked Obfuscation: Real trace, obfuscating trace and the predicted trace. The solid lines indicate Alices real traces, and the dotted lines indicate the fake traces. In (a), using machine learning, Alice's real trace and fake trace $F1$ (obfuscating trace) can be distinguished based on her move patterns. In (b), Alice generates a fake trace $F2$ (predicted trace) to approximate her real trace – in this case, $F1$ and $F2$ are indistinguishable.

VI. Obfuscation Generation

One of the most significant challenges in the query obfuscation based techniques is how to generate fake traces "alike" enough with the real user move patterns. Peddinti et al. [11] present two types of attacks depending upon whether a short-term query history is available. When history is available, a Classification Attack can be performed by using machine learning such as the Support Vector Machine (SVM) classifiers which can be trained with the user training data and the fake query training data generated from known user trips. In the absence of history, a Trip Correlation Attack can be performed based on two metrics C distance and average speed. For the security parameter $k = 5$, i.e., 4 fake traces will be generated to obfuscate one real trace, the classification attacks can identify up to 93.67% of user trips, with only 2.02% of fake trips misclassified; and the trip correlation attacks can increase the user query identification probability from 20% to 40%.

We address the trace identification problem from a new perspective: instead of generating fake traces resemble to the real ones, we involve obfuscation in real user move patterns to make it resemble to the fake ones, while still guarantee the accuracy of the location information. We name this method "cloaked obfuscation".



(a) (b)

Figure 4: Features of the cabspotting data set: (a) report distribution over 24 days, (b) report distribution in a day.

To make the real user move patterns resemble to the fake ones, we generate the predicted trace $\mathcal{T}_P$ to approximate the real trace. As shown in Fig. 3, the solid line represents Alice's real trace, and the dotted line represents Alice's fake trace. In Fig. 3a, using trace identification, the real trace and the fake trace can be distinguished. Figure. 3b shows how this problem is solved by CBA. Every time Alice accesses LBSs in a CBA zone, she sends only fake queries with her fake locations (represented as $F1$ and $F2$). Since $F1$ and $F2$ are generated with the same algorithm, with no other background knowledge, they cannot be distinguished using machine learning.

VII. Evaluations

We now move to evaluate the performance of CBA. We use real taxi cab traces [21] which are collected in the cabspotting project [22] to track the locations of 536 cabs in the San Francisco Bay area using on-board GPS devices for 2,071,530 seconds (i.e., 24 days approximately). Each cab reports its location to the server at various time intervals (i.e., 10 second, 8 minute, 1 hour, etc), and generates 11,219,955 records in total. The distribution of the time intervals based on collection frequency is shown in Fig. 4.

In the following sections, we first evaluate the system performance, we then analyze the privacy performance of CBA under various attacks.

A. Parameter Setting

There are three parameters in our CBA scheme: the speed $\delta_{\mathscr{S}}$, the turning angle δ_θ of the users, and the frequency f of the queries ($\delta_{\mathscr{S}}$, δ_θ and f are mentioned in Section V).

We set these parameters as follows. Figure 5a shows the CDF of user speed for all the users, and that for only the frequent querying users. We set $\delta_{\mathscr{S}}$ to 36 km/h since most of the users drive slower than this speed. Figure 5b shows the CDF of the turning angle between any two records. We observe that it is unlikely that the users turn at an angle larger than $\pi/2$, thus we set δ_θ to $\pi/2$. Figure 5c shows the CDF of query intervals. We set f to 1/60 Hz since 80% percent of the queries are sent within 60 seconds.

B. System Performance

Among the existing techniques, two approaches – Path Confusion [8] and Query Obfuscation [19] – can protect privacy while providing QoS guarantee. We compare the performance of CBA with these two approaches, and report the results in the sections as follows.

1) Waiting Time: We first evaluate CBA in terms of *waiting time*, and compare its performance with Path Confusion. We use *waiting time* t_w to indicate the time interval when a user is not covered by the CBA scheme. It is defined as the time interval between a user leaving a CBA zone and joining another CBA zone.

We compare the *waiting time* between CBA and Path Confusion. The results in Figures 6a and 6b show that CBA increases the chance for a user to join an anonymity group by about 10 times. Figure 6a shows that on average a user in CBA creates a CBA zone every 10 minutes, as compared to more than two hours in Path Confusion. Figure 6b shows that even in the worst case a user is still able to create a CBA zone every 3 hours, and this is in comparison to more than a day in Path Confusion.

2) Cost-Effectiveness Ratio: To indicate the effectiveness of resources used in CBA, we define *cost-effectiveness ratio* as follows.

$$r = \frac{k}{\sum_u \mathcal{W}_{\mathcal{T}} / \sum_u \mathcal{W}_{\mathcal{T}_R}}$$

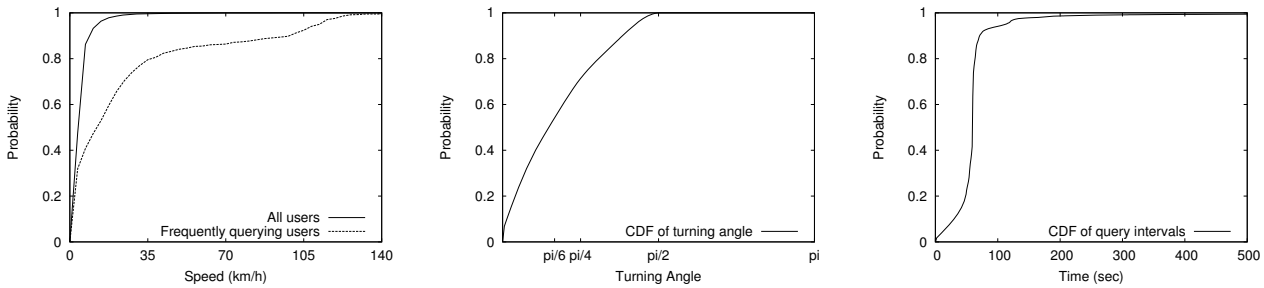
where k represents the anonymity degree (the k-anonymity parameter), and $\mathcal{W}$ represents the resource consumed (quantified as the number of queries), $\mathcal{T}$ represents a trace generated by user u, and $\mathcal{T}_R$ represents his real trace. To achieve the same anonymity degree, less resources used results in a higher *cost-effectiveness ratio*.

In this experiment, we compare CBA with Query Obfuscation in terms of r, as shown in Fig. 6c. From the figure, we observe that the *cost-effectiveness ratio* of CBA is much higher than that of Query Obfuscation. For Query Obfuscation, the *cost-effectiveness ratio* is constant, i.e., $r_{QueryObfucation} = 1$. For CBA, fake queries are generated only in CBA zones, thus only a small proportion (i.e., $1/m$, where $m > 1$) of queries are fake queries. The *cost-effectiveness ratio* for this user can be calculated as follows.

$$r_{CBAzone} = \frac{k}{\mathcal{W}_{\mathcal{T}_O} + \mathcal{W}_{\mathcal{T}_P}/\mathcal{W}_{\mathcal{T}_P}} = \frac{k}{1 + 1/m} = \frac{m \times k}{m + 1}.$$

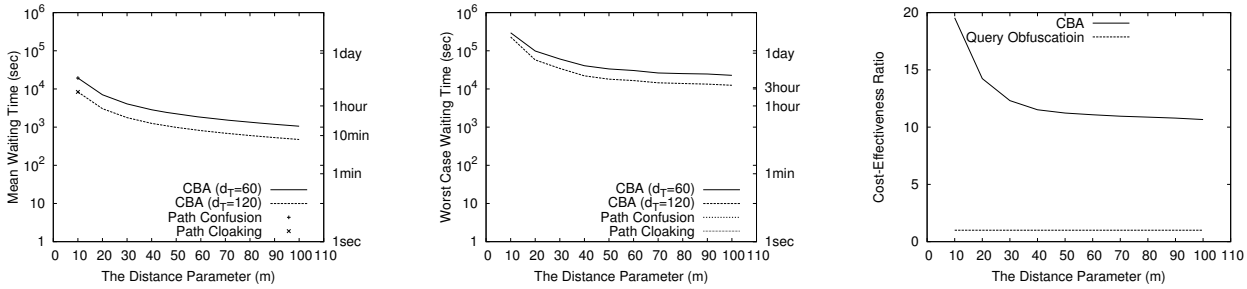
Figure 6c also shows that the *cost-effectiveness ratio* decreases with a larger CBA zone, due to more fake queries generated. We find that even when the diameter of a CBA zone is up to 100 meters, the *cost-effectiveness ratio* of CBA still achieves 10, i.e., to achieve the same anonymity degree, CBA only consumes at most 10% of the resources consumed by Query Obfuscation.

As shown in Fig. 6, there exists a tradeoff between the *waiting time* and the *cost-effectiveness ratio*. According to our experimental results, we find that setting the spatial



(a) CDF of user speed (b) CDF of turning angles (c) CDF of report intervals

Figure 5: Parameter setting: (a) setting the speed $\delta_{\mathscr{S}}$, (b) setting the turning angle δ_θ, (c) setting the frequency f.



(a) (b) (c)

Figure 6: System performance comparison: (a) comparison of mean *waiting time* between CBA and Path Confusion, (b) comparison of worst case *waiting time* between CBA and Path Confusion, (c) comparison of the *cost-effectiveness ratio* between CBA and Query Obfuscation.

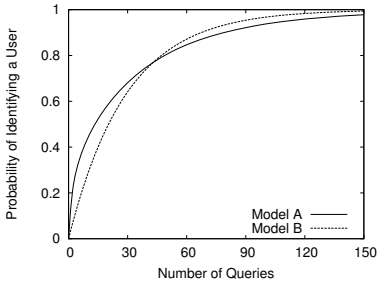


Figure 7: Identification attacker model

distance parameter d_S (the upper bound of the diameter of CBA zones) to 100 meters achieves the best tradeoff.

C. Privacy Performance

We now evaluate the privacy performance of CBA under two kinds of attacks: the identification attack and the classification attack.

1) Identification Attack: With the knowledge of user queries, an attacker may mount an identification attack to identify a target. In this case, the spatial and temporal information of a query will be analyzed by an attacker to track an anonymous user. If any of these locations in the trace can be linked with a certain identity, the attacker knows with high confidence this user's real identity.

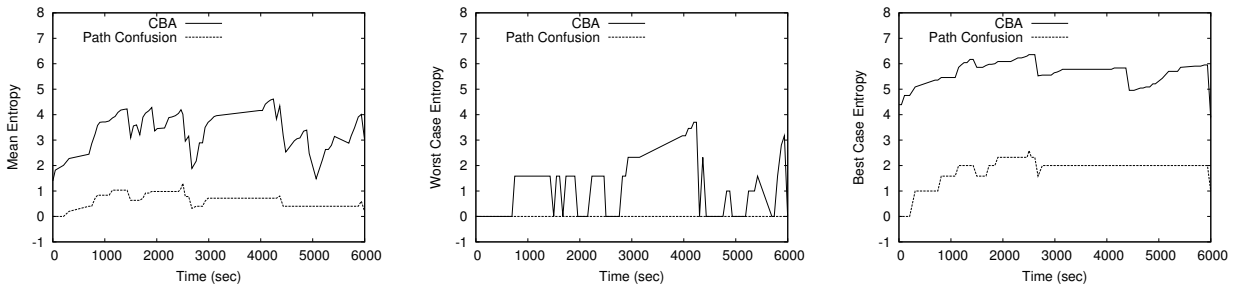
Attacker Model: We introduce two models to simulate how an attacker identifies a user.

$\mathcal{A}$. **Designation Model.** Every cab occasionally stops querying or stops moving at different locations for various periods of time. We treat these locations "identifying locations".

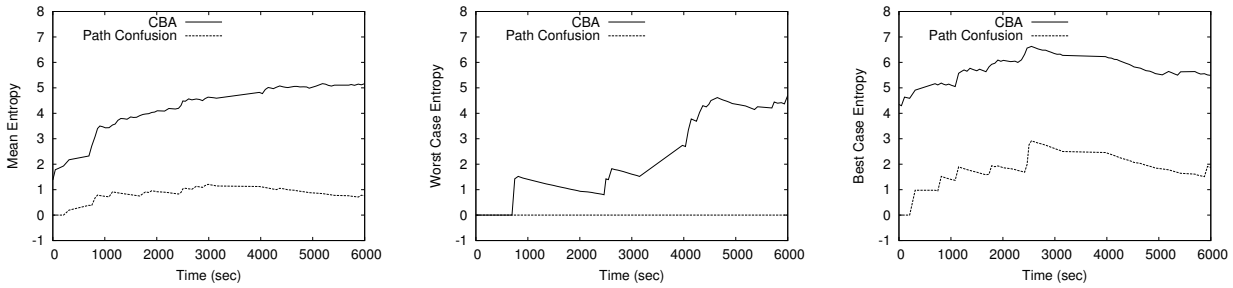
$\mathcal{B}$. **Iteration Model.** We use an even distribution model to simulate the trend that the probability of identifying a user is increased with the number of queries sent by a user.

Privacy Metrics: Path Confusion uses a threshold $\mathcal{H} = -\sum p_i \cdot \log_2 p_i$ to define tracking uncertainty, where p_i denotes the probability that location sample i belongs to the vehicle currently tracked. Lower values of H indicate more certainty or lower privacy [23]. For each identifying location model proposed, we measure the degree of privacy as the time that an attacker can correctly follow a trace, i.e., the trend that the attacker's uncertainty $\mathcal{H}$ goes with the user's online time t.

Results and Analysis: For each user in the data set, we calculate the location entropy using both attacker models. The results comparing CBA with Path Confusion are shown



(a) Mean entropy over time in model A (c) Worst case entropy in model A (e) Best case entropy in model A



(b) Mean entropy over time in model B (d) Worst case entropy in model B (f) Best case entropy in model B

Figure 8: Privacy entropy in different cases in both models: comparison between CBA and Path Confusion

in Fig. 8. In all the cases, CBA has much better privacy performance than Path Confusion.

2) Classification Attack: As discussed in Section VI, under an classification attack, an attacker distinguishes fake traces from user move patterns. We now demonstrate that the security of CBA under classification attacks reduces to that of Mix Zone.

Security Assumption of Identifying Mix Zone: Let $u, v \in \mathcal{U}$ be chosen at random in a Mix Zone $\mathcal{Z}$, $\mathcal{T}_1$, $\mathcal{T}_2$ be the traces of them after they pass $\mathcal{Z}$. We define the security assumption of identifying Mix Zone as that no machine learning algorithm $\mathcal{M}$ can distinguish u's trace $\mathcal{T}_u$ from $\mathcal{T}_1$ and $\mathcal{T}_2$ with more than a negligible advantage. The advantage of $\mathcal{M}$ is $|P(\mathcal{T}_u = \mathcal{T}_1) - P(\mathcal{T}_u = \mathcal{T}_2)|$.

Attacker Model: To simulate a classification attack, suppose an attacker knows all the information the LSP knows, the attacker may generate fake queries based on historical information. Then he chooses a classifier and trains the classifier with both the user data and the fake query data. If a classifier C has an classification accuracy $\mathcal{P}_C$, the probability of distinguishing a fake trace in a CBA zone is $\mathcal{P}_C$.

Privacy Analysis: We now conduct experiments to analyze if CBA is security under classification attack. We use various classifiers available in weka, such as naive bayes, support vector machines, AD trees, J48 trees, etc. Using these classifiers, the probability of identifying the real traces ranges from 0.5 to 0.57 when k = 2. To this aspect, CBA is secure. Proving the security of CBA is constrained by the classifier used, instead we prove that the security of CBA in the above attacker model can be reduced to the hardness of the security assumption of identifying Mix Zone.

Proof of Security: Suppose there exists a machine learning algorithm $\mathcal{N}$ that can distinguish users u, v who pass the same CBA zone with more than negligible advantage σ. For the same users u, v, assume the CBA zone to be a Mix Zone, we have their traces before they enter this Mix Zone and the traces after they leave this Mix Zone. We use the trace generation algorithm $\mathcal{F}_{TraceGen}$ (as in Section V) to generate four traces connecting their entries and their exits to this Mix Zone. According to Definition 1, the Mix Zone with these four traces is a CBA zone. Then we have composition algorithm $\mathcal{M} = C_{\mathcal{F}_{TraceGen}}\mathcal{N}$, which can distinguish u's trace $\mathcal{T}_u$ from $\mathcal{T}_1$ and $\mathcal{T}_2$ with more than a negligible advantage σ.

Discussion on Location Accuracy: Querying LBSs with only fake locations decreases the QoS. We conduct experiments on this factor, and the results are shown in Fig. 9. We observe that even in the worst case the distance between the predicted queries and the real queries is less than 10 meters, which is similar to the accuracy provided by GPS. We believe that the price is reasonable to tradeoff privacy.

VIII. Conclusion

This paper presents a novel CBA scheme to balance user privacy and QoS for location-based services. In CBA, we propose the collaborative path confusion mechanism for nearby users to generate a CBA zone, the local prediction

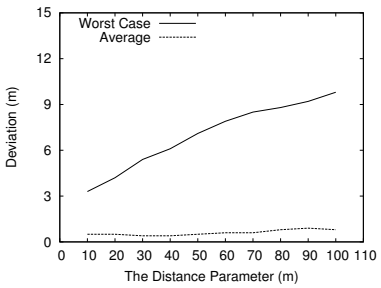


Figure 9: The distance between the predicted queries and the real queries.

algorithm for each user to generate obfuscating traces for path confusion, and the cloaked obfuscation method to prevent the trace identification attacks. Using a real-world data set, we demonstrate that CBA outperforms Path Confusion and Query Obfuscation in terms of robustness and resources consumed.

Acknowledgment

This work was supported by Natural Science Foundation of China (NSFC) under grant 61073031, National 973 Program of China under grant 2009CB320702, and Innovation Research Group Program of NSFC under grant 61021062.

References

[1] D. Goldschlag, M. Reed, and P. Syverson, "Onion routing," *Communications of the ACM*, vol. 42, no. 2, pp. 39–41, 1999.

[2] M. Gruteser and D. Grunwald, "Anonymous usage of location-based services through spatial and temporal cloaking," in *Proceedings of the 1st international conference on Mobile systems, applications and services*, 2003.

[3] J. Meyerowitz, R. Choudhury *et al.*, "Hiding stars with fireworks: location privacy through camouflage," in *Proceedings of the 15th annual international conference on Mobile computing and networking*, 2009.

[4] B. Gedik and L. Liu, "Location privacy in mobile systems: A personalized anonymization model," in *Proceedings of the 25th IEEE International Conference on Distributed Computing Systems*, 2005.

[5] M. Mokbel, C. Chow, and W. Aref, "The new Casper: query processing for location services without compromising privacy," in *Proceedings of the 32nd international conference on Very Large Data Bases*, 2006.

[6] P. Kalnis, G. Ghinita, K. Mouratidis, and D. Papadias, "Preventing location-based identity inference in anonymous spatial queries," *IEEE Transactions on Knowledge and Data Engineering*, vol. 19, no. 12, pp. 1719–1733, 2007.

[7] A. Beresford and F. Stajano, "Location privacy in pervasive computing," *Pervasive Computing*, pp. 46–55, 2003.

[8] B. Hoh and M. Gruteser, "Protecting location privacy through path confusion," in *Security and Privacy for Emerging Areas in Communications Networks, 2005. SecureComm 2005. First International Conference on*. IEEE, 2005, pp. 194–205.

[9] B. Hoh, M. Gruteser, H. Xiong, and A. Alrabady, "Preserving privacy in gps traces via uncertainty-aware path cloaking," in *Proceedings of the 14th ACM conference on Computer and communications security*, 2007.

[10] B. Palanisamy and L. Liu, "Mobimix: Protecting location privacy with mix-zones over road networks," in *Data Engineering (ICDE), 2011 IEEE 27th International Conference on*, 2011.

[11] S. Peddinti, N. Saxena, and A. Birmingham, "On the limitations of query obfuscation techniques for location privacy," in *International conference on Ubiquitous computing*, 2011.

[12] L. Sweeney, "Achieving k-anonymity privacy protection using generalization and suppression," *International Journal of Uncertainty Fuzziness and Knowledge-Based Systems*, vol. 10, no. 5, pp. 571–588, 2002.

[13] A. Kapadia, N. Triandopoulos, C. Cornelius, D. Peebles, and D. Kotz, "AnonySense: Opportunistic and privacy-preserving context collection," *Pervasive Computing*, pp. 280–297, 2008.

[14] T. Hashem and L. Kulik, "dont trust anyone: Privacy protection for location-based services," *Pervasive and Mobile Computing*, 2011.

[15] C. Chow, M. Mokbel, and X. Liu, "A peer-to-peer spatial cloaking algorithm for anonymous location-based service," in *Proceedings of the 14th annual ACM international symposium on Advances in geographic information systems*, 2006.

[16] G. Ghinita, P. Kalnis, and S. Skiadopoulos, "PRIVE: anonymous location-based queries in distributed mobile systems," in *Proceedings of the 16th international conference on World Wide Web*, 2007.

[17] H. Kido, Y. Yanagisawa, and T. Satoh, "An anonymous communication technique using dummies for location-based services," in *Proceedings of the 2nd International Conference on Pervasive Services, 2005.* IEEE, 2005, pp. 88–97.

[18] J. Krumm, "Realistic driving trips for location privacy," *Pervasive Computing*, pp. 25–41, 2009.

[19] P. Shankar, V. Ganapathy, and L. Iftode, "Privately querying location-based services with sybilquery," in *Proceedings of the 11th International Conference on Ubiquitous Computing*, 2009.

[20] "Microsoft multimap api," http://classic.multimap.com/.

[21] M. Piorkowski, N. Sarafijanovoc-Djukic, and M. Grossglauser, "A Parsimonious Model of Mobile Partitioned Networks with Clustering," in *The First International Conference on COMmunication Systems and NETworkS (COMSNETS)*, January 2009. [Online]. Available: http://www.comsnets.org

[22] "Cabspotting," http://cabspotting.org/.

[23] C. Shannon, "A mathematical theory of communication," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 5, no. 1, pp. 3–55, 2001.