2009 29th IEEE International Conference on Distributed Computing Systems

CAP: A *C*ontext-*A*ware *P*rivacy Protection System for Location-Based Services

Aniket Pingley, *George Washington Univ.*, *apingley@gwu.edu*
Wei Yu, *Cisco Systems*, *weyu@cisco.com*
Nan Zhang, *George Washington Univ.*, *nzhang10@gwu.edu*
Xinwen Fu, *UMass Lowell*, *xinwenfu@cs.uml.edu*
Wei Zhao, *Rensselaer Polytechnic Institute*, *zhaow3@rpi.edu*

Abstract

We address issues related to privacy protection in location-based services (LBS). Most existing research in this field either requires a trusted third-party (anonymizer) or uses oblivious protocols that are computationally and communicationally expensive. Our design of privacy-preserving techniques is principled on not requiring a trusted third-party while being highly efficient in terms of time and space complexities. The problem has two interesting and challenging characteristics: First, the degree of privacy protection and LBS accuracy depends on the context, such as population and road density, around a user's location. Second, an adversary may violate a user's location privacy in two ways: (i) based on the user's location information contained in the LBS query payload, and (ii) by inferring a user's geographical location based on its device's IP address. To address these challenges, we introduce CAP, a C*ontext-*A*ware* P*rivacy-preserving LBS system with integrated protection for data privacy and communication anonymity. We have implemented CAP and integrated it with Google Maps, a popular LBS system. Theoretical analysis and experimental results validate CAP's effectiveness on privacy protection, LBS accuracy, and communication Quality-of-Service.*

1. Introduction

Location-based service (LBS) provides a user with contents customized by the user's current location, such as the nearest restaurants/hotels/clinics, which are retrieved from a spatial database stored remotely in the LBS server. LBS not only serves individual mobile users, but also plays an important role in public safety, transportation, emergency response, and disaster management. With an increasing number of mobile devices featuring built-in Global Positioning System (GPS) technology, LBS has experienced rapid growth in the past few years. According to the ABI research report [19], the number of GPS-enabled LBS subscribers is projected to reach 315 million by 2013.

A request for LBS can be considered a query over the LBS server's spatial database. For example, a query for the ten nearest four-star hotels can be expressed as the following SQL-like top-k query:

SELECT TOP 10 FROM *Hotel*
WHERE STARRATING = 4
ORDER BY DISTANCE(*Hotel.Location*, *userLoc*) ASC;

where *userLoc* is the user's location. Note that the user's location is specified as a constant in the ranking function and should be sent along with the query to the LBS server.

Despite the benefits provided by LBS, a user may not be willing to provide its current location to the LBS server due to concerns on location privacy. Such concerns can be attributed to the seriousness of location disclosure and misuse: For example, an adversary may learn a user's political and religious affiliations based on the locations the user regularly visits. In recent years, there have been several reports on the abuse of LBS by individuals and companies to intrude others' privacy [16].

The objective of a privacy-preserving LBS is to protect the privacy of a user's location while maintaining a high level of LBS accuracy (e.g., the rank of a 4-star hotel in the above example). It has received growing attention from the research community. A k-anonymity based framework was proposed to protect location privacy by using a trusted third-party called the anonymizer [11]. With this framework, a user sends its location to the centralized anonymizer, which subsequently generates a k-anonymized [22] *cloaking region* that covers not only this user, but also $k-1$ other users. Then, the anonymizer transmits the cloaking region to the LBS server as the constant in the LBS query, and forwards the query answer to the user. This framework prevents the LBS server from distinguishing a user among at least $k-1$ others.

Unfortunately, in real systems, it may be difficult, if not impossible, to find a trusted third-party anonymizer, especially one which has a large user base to shrink the cloaking region for better LBS privacy. To the best of our knowledge, the only existing work which removes the requirement of a trusted third-party is a private information retrieval (PIR)-based approach [7]. Nonetheless, this approach has two critical drawbacks. First, it can only be applied to LBS servers which support the PIR-based protocol. Second, as a common problem for PIR-based techniques, it may incur high computational and communication overhead unaffordable to mobile devices and the LBS server[1].

1. It was shown that PIR may incur even higher communication overhead than an oblivious transfer of the entire server-side database [21]. Such a cost may be prohibitive for the LBS server if it needs to process concurrently a large number of LBS queries.

1063-6927/09 $25.00 © 2009 IEEE
DOI 10.1109/ICDCS.2009.62

IEEE computer society

In this paper, we initiate the investigation of a privacy-preserving technique that is efficient in terms of both time and space complexities, does not require a trusted third-party, and is transparent to the LBS server so that it can be readily deployed with existing LBS systems. Such a technique may have to make a tradeoff between privacy protection and LBS accuracy. Nonetheless, it should provide effective guarantees on both measures.

A straightforward method for efficient privacy protection is to randomly perturb a user's location based on pre-determined noise distributions on longitude and latitude. This method is, in principle, similar to the randomization approach for privacy-preserving data mining [24]. Nonetheless, it is unlikely to suffice for LBS because, with a pre-determined noise distribution, the levels of privacy protection and LBS accuracy largely depend on the "context", such as road and population density, around a user's location. For example, intuition suggests that, to achieve the same level of privacy and LBS accuracy, a user should (or could) deviate more from its real location in a rural area than in downtown.

Thus, a critical challenge for privacy-preserving LBS is to achieve context-aware privacy protection. The existing k-anonymity framework does so by leveraging the anonymizer's global knowledge of user distribution (so that the cloaking region is automatically larger in a rural area which has fewer users). Without a trusted third-party, we must acquire the context information from other sources. A simple solution is for each mobile device to store a complete topology map and retrieve it before perturbation to compute the adjacent area's context. However, this may lead to computational and storage overhead unaffordable to mobile devices that are not designated GPS navigation systems.

In this paper, we introduce *CAP*, a *C*ontext-*A*ware *P*rivacy-preserving LBS system. The main idea behind CAP is a dimension-reducing projection of every 2-d geographical location to a 1-d space, such that (i) every point in the 1-d space has homogeneous context (e.g., equal road/population density), and (ii) adjacent locations remain close after the projection. We refer to such a projection as a Various-grid-length Hilbert Curve (VHC)-mapping. With CAP, a user first projects its current location to the 1-d space based on VHC-mapping, and then randomly perturbs the 1-d value based on a pre-determined noise distribution. The perturbed value is mapped back to the 2-d space according to VHC-mapping and then transmitted as the user's location to the LBS server.

VHC-mapping is designed to provide guarantees on both privacy protection and LBS accuracy. It is also very efficient in terms of both time and space complexities: The VHC-map itself is computed offline based on a real-world topology map, but only costs minimal storage space (e.g., our experiments use a VHC-map which is only $1/2000$ the size of a topology map) and retrieval cost. The usage of perturbation technique ensures transparency to the LBS server, and enables CAP to be readily integrated into existing LBS systems.

In the design of CAP, we also initiate an investigation of the network anonymity perspective of location privacy. Existing work has shown that a user's location may be derived from its IP address based on public information about base stations' locations and IP addresses [6]. For example, when 802.11b base stations are used, the user may be positioned within a small radius of 50 meters. As such, without a trusted third-party anonymizer, location privacy may be breached through not only an LBS query, but also the traffic that carries the query. To address this problem, we use Tor [4], a popular anonymous routing network, to hide a user's IP address. Unfortunately, we found that Tor suffers from serious Quality-of-Service (e.g., response time) degradation which may be unbearable for mobile (e.g., driving) applications that require short response time. To solve the problem, we present a set of new routing algorithms for Tor which reduce latency and maximize throughput.

To the best of our knowledge, CAP is the first real privacy-preserving LBS system that provides an efficient and context-aware solution for both data privacy and communication anonymity without the presence of a trusted third-party. We have implemented CAP in both SUSE Linux 11.0 and Mac OSX Operating Systems, and are porting the system to Linux and OSX-based mobile devices. More information about the system implementation can be found at *http://seas.gwu.edu/∼nzhang10/cap*.

The remainder of the paper is organized as follows. In Section II, we formally specify the problem and present the architecture of CAP. Section III is devoted to the development of VHC-mapping. In Section IV, we discuss other design issues of CAP, including the anonymous routing. Section V contains a detailed experimental evaluation of CAP. Section VI discusses the related work. We conclude in Section VII.

2. System Overview of CAP

In this section, we present an overview of CAP, our context-aware privacy-preserving LBS system. The focus is on the system infrastructure of CAP and its performance measures.

2.1. Parties

There are two parties in the system: a user who uses the LBS and a server which provides it. In practice, a user may be a mobile device, such as a laptop, PDA, cell phone, etc, which obtains its location from a positioning device such as a GPS receiver. Examples of LBS server include point-of-interest search engines such as Google Maps (*http://maps.google.com*).

The interactions between the two parties can be stated as follows: The user issues an LBS query to the server. The LBS query is a top-k query with ranking function specified as the distance to the user's current location. After

receiving the LBS query, the server executes it against a spatial database and returns the answer to the user.

Due to privacy concerns, the user is unwilling to disclose its location to the server. Thus, the user's objective is to obtain the relatively accurate LBS query answer without disclosing its real location. The server is supposed to correctly answer the received LBS query. Besides, the objective of a malicious server is to compromise the user's location. In this paper, we refer to a malicious server as an adversary.

2.2. System Architecture

Figure 1 illustrates the baseline architecture of CAP. Recall that there are two possible ways for a user's location to be disclosed: through the location information included in the LBS query, or through the user's network (e.g., IP) address. CAP has two components, location perturbing and anonymous routing, principled on eliminating these two disclosure channels, respectively.

The *location perturbing component* perturbs the user's location included in the LBS query. It also rearranges the results returned by the LBS server based on the original user location, in order to provide better data utility. The *anonymous routing component* hides the user's network identity by routing the LBS query through relaying nodes in an anonymous communication network, Tor, before sending it to the LBS server.

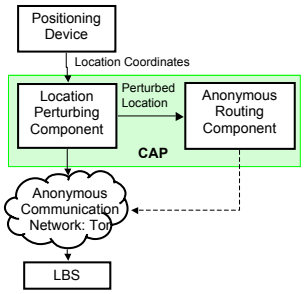

Figure 1. Baseline Architecture of CAP

2.3. Performance Measures

The performance of a privacy-preserving LBS system should be measured in terms of the accuracy of LBS query answer, the privacy protection of user's location, and the communication quality-of-service (e.g., query response time). We define these three measures respectively, as follows.

2.3.1. Accuracy Measure. Since an LBS query is essentially a top-k query over a spatial database, we consider accuracy measures for top-k queries. A number of measures have been proposed, including rank distance (i.e., the difference between the returned and the true rank of a returned tuple), true positive rate (i.e., the probability that a tuple in the result is indeed a true top-k tuple), score distance (e.g., the extra distance driven according to the returned tuples), etc [1]. In the theoretical analysis part of this paper, we adopt rank distance as the accuracy measure. Nonetheless, in the experimental results, we shall evaluate other possible measures such as the true positive rate.

Definition 2.1. *The average rank distance of a privacy-preserving scheme that perturbs userPos from x to $R(x)$ is*

$$l_{\mathrm{r}}(x) = AVG_{t \in q(x)}(|rank(t, q(x)) - rank(t, q(R(x))|).$$

where $AVG(\cdot)$ represents the average value, $q(x)$ is the LBS query answer when userPos $= x$, and $rank(t, q(x))$ is the rank of tuple t in the returned answer $q(x)$.

2.3.2. Privacy Measure. Our privacy measure is principled on the same anonymity standard as k-anonymity. The difference, however, is that our system does not feature a trusted third-party which has a global view of active users. Thus, our measure is defined over the population among which the user is hidden. This is similar to the usage of historic footprints of active users for the k-anonymity definition in [23].

Definition 2.2. *A privacy-preserving scheme which perturbs userPos from x to $R(x)$ satisfies N-camouflage iff there exists a region C of population at least N, such that for any subregion $C' \subseteq C$, $\Pr\{x \in C' | R(x)\} = |C'|/|C|$, where $|\cdot|$ is the area of the region.*

According to the definition, a privacy-preserving scheme satisfies N-confidentiality iff no adversary can distinguish between any two locations in a region of population N.

2.3.3. QoS Measure. Since an LBS user may be constantly moving, the overhead of LBS query processing is important for the utility of LBS. Such an overhead is a combination of three parts: the location perturbing component, the random routing protocol of anonymous routing network, and the query processing at the LBS server. Since CAP is transparent to the LBS server, we discuss the first two parts in the paper.

3. Location Perturbing Based on VHC-Mapping

We focus on the location perturbing component of CAP in this section. We begin with introducing our basic ideas, and then substantiate the ideas by describing VHC-mapping, our main technique for this component.

3.1. Key Idea

Recall that the location perturbing component perturbs a user's position included in an LBS query before sending the query to the LBS server. The objective is to provide "context-aware" perturbation without incurring the cost of storing and retrieving a full-scale topology map in a mobile device. Our key idea is to pre-compute a projection from

the original space (of latitude and longitude) to a new space, such that

- the projection is locality-preserving i.e., two nearby points in the original space are also close in the projected space, and vice-versa,
- all points in the new space have homogenous "context" i.e., population density, and
- the projection must be stored with space orders of magnitude smaller than the topology map, and can be efficiently computed.

After projecting a user's location to the new space, we apply homogeneous perturbation to all mapped points in the new space, project the perturbed points back to the original 2-d space, and then output the result as the perturbed location.

Figure 2(a) provides a simple illustration of the projection on 1-d data, where the population density is defined based on 6 people A to F. In the original space, the population density near B, C, or D is higher than A, E, or F. The mapping is designed such that every point in the new space has equal density. Thus, the same noise applied to B, C, or D will become smaller after being mapped back to the original space. This is consistent with our intuition that, in order to provide universal privacy and accuracy guarantees for all locations, less perturbation should be applied a higher-density area.

3.2. VHC-Mapping

We now introduce *V*arious-size-grid *H*ilbert *C*urve (VHC)-mapping, our main technique for the projection to homogeneous-context space. We will first describe the construction of VHC-mapping, and then discuss how it satisfies the above-mentioned three conditions.

3.2.1. Construction of VHC-mapping. The construction of VHC-mapping must refer to context information such as road or population density. In the design of CAP, we choose road density as input because (i) economic studies show that road and population densities are strongly correlated, following (approximately) a linear relationship [8], and (ii) in practice, road density information is readily available[2] and usually more accurate than population information. Nonetheless, our design of VHC-mapping can be easily adapted to population density.

Without loss of generality, we consider the original 2-d latitude/longitude space as a square. VHC-mapping involves a recursive partitioning of the square into various-size cells according to context information. Each cell is either partitioned into 4 equal-size square cells, or not (further) partitioned (i.e., becomes a base cell), based on the following rule:

2. To calculate the road density of an area, we use the information provided by the by the US Census Bureau Topological Integrated Geographic Encoding and Referencing (TIGER) system which contains information about roads for every county in the US.

Min-Density Rule: *Partition a cell into 4 equal-size subcells iff the total road length (in the original space) covered by the cell is at least μ times the edge length of the cell, where $\mu > 1$ is a pre-determined granularity ratio.*

An example of the partitioning result is shown in Figure 2(b). One can see that the base cells have three possible sizes. According to the min-density rule, a larger base cell represents an area with lower road density.

After the partitioning process, we construct the mapped 1-d space as a variation of the Hilbert space-filling curve [17] to connect all various-size cells in the original 2-d space. Figure 2(b) depicts an example of such a Hilbert curve, while Figure 2(c) demonstrates a real implementation on the map of Baltimore, MD with granularity ratio $\mu = 20$.

The VHC-mapping is then constructed as follows: A 2-d point in the original space is mapped to its (geographically) nearest point on the Hilbert curve. A 1-d point, after being perturbed by additive noise, is mapped back to the original space by randomly selecting a 2-d point which can be mapped to the 1-d perturbed point.

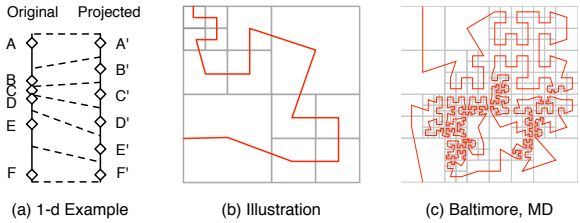

Figure 2. Examples of VHC-Mapping

3.2.2. Justification. We now explain how VHC-mapping satisfies the three requirements we outlined in Section 3.1: (i) locality-preserving, (ii) constant density, and (iii) efficiency of storage and retrieval.

First, a well-known property of Hilbert curve is locality preserving, e.g., two adjacent points in the projected space are likely to be close in the original space. Thus, VHC-mapping satisfies the locality-preserving requirement.

Next, for the constant-density requirement, there are two key observations: First, due to the min-density rule, the total road length covered by each base cell is at most μ times the edge length of the cell. Second, due to our construction of the VHC, the length of the Hilbert curve covered by a base cell is approximately the same as the edge length of the cell.

As such, intuitively, every point on the Hilbert curve (i.e., in the projected space) can be considered as corresponding to about μ points *on the roads* in the original space. Thus, the road density is approximately constant for all points in the projected space. This fulfills the constant-density requirement.

We now consider the third requirement on the efficiency of storing and conducting VHC-mapping. VHC-mapping can be stored as a 4-tree based on the partitioning of the original space, where each node is either a leaf node (if

corresponding to a base cell) or has 4 children (if further partitioned). Figure 3(b) depicts an example of such a 4-tree for the VHC-mapping in Figure 3(a). One can see from the figure that base cells of different sizes are corresponding to leaf nodes at different layers of the tree.

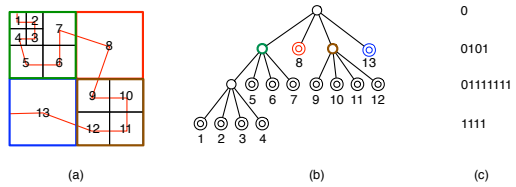

Figure 3. 4-Tree for Storage of VHC-Mapping

Since each node either is a leaf node or has 4 children, we only need to store 1-bit information to indicate whether it is a leaf. Figure 3(c) shows an example of such encoding scheme for the tree in Figure 3(b). Since a 4-tree with n leaf nodes has at most $4n/3$ (total) nodes, the space required by the serialized map file is at most $4n/3$ bits.

Based on the 4-tree, VHC-mapping can be retrieved and used as follows: First, we reconstruct the 4-tree from the serialized map file and traverse every leaf node to assign its corresponding range in the 1-d projected space. In particular, a leaf node at level i is corresponding to a range of length $d/2^i$ where d is the edge length of the entire map. This step has time complexity of $O(n)$. Then, we can conduct VHC-mapping by searching for the corresponding leaf node (i.e., base cell) of the original 2-d location. The time complexity is $O(\log n)$. The inverse mapping of a 1-d location in the projection space back to the original space can be done through a binary search on all leaf nodes. The time complexity is $O(\log n)$.

3.3. Algorithms for VHC-Mapping

We now present two detailed algorithms for our approach: One is the offline construction and storage of VHC-mapping. The other is the online retrieval of VHC-mapping and the perturbation of a user's locations.

Algorithm VHC-Build: Offline Construction

Require: Map, C as the (rectangle) boundary of the map

1: Store $C \| \text{BUILDTREE}(C)$ as the HC-mapping file.
2: **function** BUILDTREE(C)
3: **if** total road length in $C \geq \mu \cdot$ edge length of C **then**
4: Partition C equally into C_{nw}, C_{ne}, C_{se}, C_{sw}.
5: **for** $i = \text{nw}, \text{ne}, \text{se}, \text{sw}$ **do**
6: **return** $0 \| \text{BUILDTREE}(C_i)$
7: **end for**
8: **else**
9: **return** 1
10: **end if**
11: **end function**

Algorithm VHC-Build depicts the offline construction and storage of VHC-mapping. In the algorithm, we use $\|$ to represent the concatenation operation. We partition the original map based on the min-density rule (Line 3) and store the 4-tree into a bit stream $\text{BUILDTREE}(C)$ (Line 6).

Algorithm VHC-Perturb: Online Location Perturbation

Require: Pre-computed VHC-mapping file $hcFile$

1: Load a 4-tree T of the partition from $hcFile$ and assign the 1-d value range for each base cell.
2: Wait until receiving $userPos$ for perturbation.
3: Find the mapped value $F(userPos)$ based on the 1-d value range of the base cell which contains $userPos$.
4: Generate random noise r according to uniform distribution on $[-\sigma, \sigma]$.
5: Compute $R(userPos) = F^{-1}(F(userPos) + r)$ by searching for the base cell which contains 1-d value $F(userPos) + r$. Output $R(userPos)$.
6: Goto 2

Algorithm VHC-Perturb depicts the online retrieval of VHC-mapping for the perturbation of a user's location. Given a 2-d location $userPos$, we map it to 1-d point $F(userPos)$, add a homogeneous noise r, and use $F^{-1}(F(userPos) + r)$ as the perturbed location. Note that r is generated from a pre-determined distribution.

Algorithm VHC-Build is executed offline and has computational complexity of $O(n)$. The computational complexity of Algorithm VHC-Perturb is $O(n)$ for the retrieval of VHC-mapping file (i.e., Line 1) and $O(\log n)$ for the perturbation of each location.

4. Discussion

In a practical LBS, a mobile's request should be served in a timely fashion. Otherwise, it may no longer be useful when the mobile has already left the location where the request was made. Recall from Figure 1, the anonymous communication network also contributes to the overhead of LBS query processing. We now discuss how to tune up the communication QoS of the anonymous routing component.

In *CAP*, Tor [4] is used for anonymous communication between clients and servers. The challenge of tuning up Tor for an LBS system is how to optimize its QoS while preserving anonymity. Tor has suffered serious performance degradation because of its random path selection algorithms [18]. Tor is an overlay network on the Internet providing anonymous communication. Within the Tor network, to browse a web server while hiding the connection, a client chooses a series of Tor routers from the Tor router directory. The sequence of ordered Tor routers is denoted as *path*. The number of Tor routers is the *path length*. The client negotiates session keys with the chosen routers, one by one, using the *Diffie-Hellman* handshake protocol and forms a *circuit*.

The client packs application data into cells that are transmitted over the circuit. Therefore, a set of sequential TCP connections are used to relay packets from the source to the

destination. Since Tor routers use donated bandwidth from users, who may limit it using the leaky bucket mechanism, the end-to-end throughput will be limited by the bottleneck segment [13]. We found that despite Tor's weighted bandwidth path selection algorithms, there is a high probability that a node with poor bandwidth is chosen because of the existence of a large number of small-bandwidth Tor routers.

We propose differential QoS in the Tor network in order to improve QoS. The Tor network could be partitioned into classes of Tor routers with high or low donated bandwidth. Paths drawn from the class of high-bandwidth routers can provide better performance. Paths can be chosen for flow requests based on a particular flow request's priority. In this way, high priority flows (e.g., LBS query request and response) will obtain high bandwidth and low priority flows will obtain low bandwidth. So long as user's requirements can be met with differential QoS, this will make more effective use of bandwidth.

Therefore, the anonymous routing component in Figure 1 will control Tor's routing in order to achieve differential QoS for Tor clients. We have implemented the two simple path selection algorithms in favor of differential QoS in the Tor network. The first algorithm is shown in Algorithm 1, which provides the differential routing with two priorities. This algorithm can be easily extended to support priorities larger than two. To provide a better QoS for LBS, a mobile client can choose the top priority, where a user prefers a path throughput greater or equal to $MinBW$.

Algorithm 1 Differentiated Routing (*Diff*)

Require: User specified minimum path throughput $MinBW$

1: Build a pool of Tor nodes whose bandwidth is greater or equal to $MinBW$.
2: Use weighted random algorithm and build a circuit through the pool. Record used Tor nodes in existing circuits and future circuits will not use those used Tor nodes.

The actual path throughput under Algorithm 1 may be much lower than $MinBW$ because of congestion on the Internet as numerous flows share the Tor nodes worldwide. To overcome this problem, the second routing algorithm (*Diff/CA*) we propose to consider the congestion avoidance as shown in Algorithm 2. Recall that Tor can create circuits proactively and wait for user connections. To avoid congestion, Diff/CA creates circuits proactively, measuring the path throughput until it meets bandwidth requirement. This incurs a delay in circuit creation. Our experiments show that the delay is within a reasonable range.

5. Experimental Results

In this section, we present the implementation and experimental evaluation of CAP. We will first introduce the implementation and the experimental setup, and then present the results for the location perturbing and anonymous routing components, respectively.

Algorithm 2 Differentiated Routing with Congestion Avoidance (*Diff/CA*)

Require: User specified minimum path throughput capacity $MinBW$ and tolerable throughput $TolBW$.

1: Build a pool of Tor nodes whose bandwidth is greater or equal to $MinBW$.
2: Use weighted random algorithm and build a circuit through the pool. Measure the circuit throughput until its bandwidth is greater or equal to $TolBW$.

5.1. Experimental Setup

We have implemented a prototypical CAP system for Mac OS X and Linux operating system with support for GPS and integration with Tor. The positioning device we used is a SiRF Star III GPS receiver which is connected to the laptop via USB interface [3]. The location perturbation component of CAP was implemented using C++ and the Boost library. Qt library and Google Maps APIs (*http://code.google.com/apis/maps/*) were used for GUI development to demonstrate the integration of CAP with existing LBS systems. For the anonymous routing component, we revised Tor version 0.1.1.26. The mobile client is connected to the Internet via 802.11b protocol. The LBS server is running on a desktop machine with 3.2Ghz Intel Core Duo CPU, 3GB RAM, and Suse 10.3 operating system.

We performed our experiments on the map of Middlesex county, Massachusetts, USA. The map was retrieved from the 2006 second edition of the Topological Integrated Geographic Encoding and Referencing (TIGER) system published by the US Census Bureau. The map can be downloaded as a zipped TIGER/Line file from *http://www2.census.gov/geo/tiger/tiger2006se/MA*.

We downloaded 800 POIs, including restaurants, hotels, clinics, and supermarkets in the county from *http://www.gps-data-team.com/poi/*. We randomly selected 1000 different co-ordinate points (latitude and longitude), lying in areas with varying road densities (e.g., downtown, rural areas, suburbs etc.), as possible user locations.

5.2. Evaluation of Location Perturbing Component

Recall that the "Online Location Perturbation" algorithm uses random noise generated from uniform distribution $[-\sigma, \sigma]$. We have tested the performance of location perturbing component by changing the noise parameter σ. We have also tested for the storage requirements by changing the granularity ratio μ (recall the "Min-Density rule" from Section 3).

To test against locations with diverse road densities, we define the *road density index* of a location as the level of the leaf node that contains this location (root has level 1). The depth of the tree is 13 when $\mu = 8$, which is used in

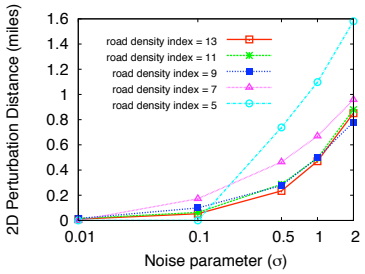

Figure 4. 2-d Perturbation distance vs σ

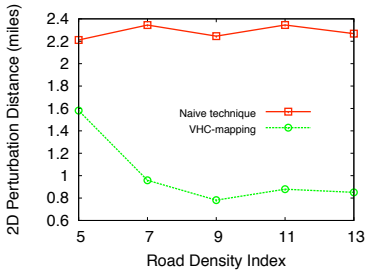

Figure 5. Naive technique vs VHC-mapping

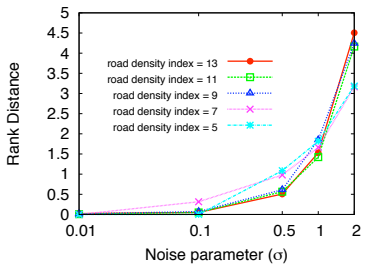

Figure 6. l_{r} (Top-10) vs. σ

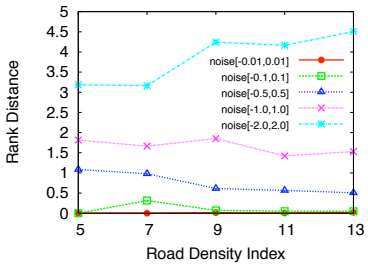

Figure 7. l_{r} (Top-10) vs. Road Density Index

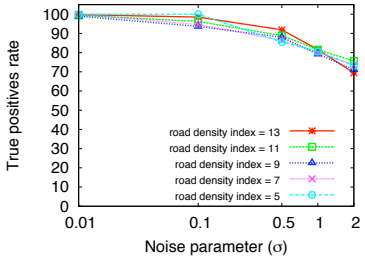

Figure 8. True Positive Rate (Top-10) vs. σ

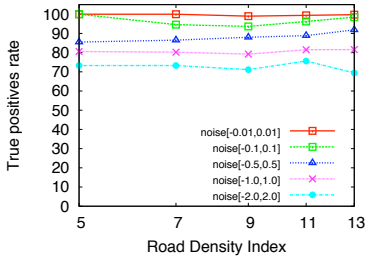

Figure 9. True Positive Rate (Top-10) vs. Road Density Index

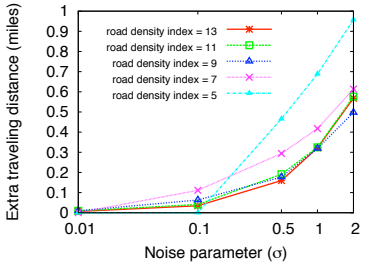

Figure 10. Extra miles vs σ

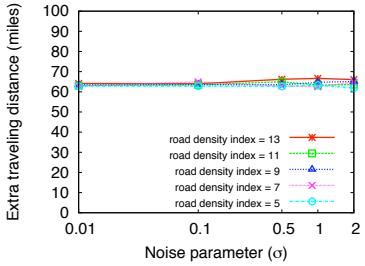

Figure 11. Extra miles (%) vs σ

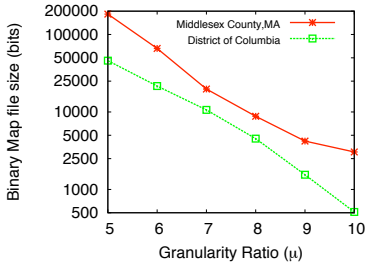

Figure 12. Binary map file size vs μ

most experiments. Generally, the road density increases in exponential order with the road density index.

Figure 4 depicts the relationship between the average 2-d perturbation distance DISTANCE($userPos$, $R(userPos)$) and the noise parameter σ for locations with various road densities. The 2-d perturbation distance is the Euclidean distance between the original and perturbed locations. Besides, we tested with Manhattan distance [12] and obtained similar results. As we can see, the 2-d perturbation distance for a rural location (road density index = 5) is much greater compared to a downtown (road density index = 13) location. This confirms that a rural location merits a larger perturbation than a downtown location.

Figure 5 depicts the comparison between VHC-mapping and a naive technique, which uses universal random noise to perturb a user's location, regardless of its context. We have used the same noise parameter value, $\sigma = 2$, for both techniques. One can clearly observe that in contrast with the naive technique, VHC-mapping applies context-aware perturbation, i.e., higher perturbation is applied to locations with lower road density.

We evaluated the accuracy of location perturbing component for a scenario where we issue a top-10 query for the nearest POI. Figures 6 and 7 depict the relationship between the degree of LBS accuracy l_{r} and the noise parameter σ for locations with various road densities. In both the figures, we can make two observations: First, l_{r} increases with the increase of σ. Second, there is no significant difference in LBS accuracy for locations with different road density indices. Similar observations can be made from Figures 8 and 9, where the LBS accuracy measure is the true positive rate of the returned top-10 results.

To estimate the real-world experience of CAP users, we consider the additional distance traveled by a user to reach

the returned nearest POI (compared with the real nearest POI). Figure 10 depicts the relationship between extra miles to be traveled and noise parameter σ. It can be observed that a user will have to travel more for higher level of privacy protection desired. However, it would be more useful to have guarantees on extra distance to be traveled given a particular level of privacy is desired. In other words, we are interested in observing the value of extra miles to be traveled as a fraction of the 2-d perturbation distance. This is depicted in Figure 11, where for all the different values of noise parameter σ, the extra miles to be traveled is approximately 65% of the 2-d perturbation distance.

Recall that the granularity ratio μ controls the size of the 4-tree (i.e., the VHC-mapping), and thus the size of binary map file. Figure 12 depicts the relationship between the storage cost of the 4-tree / binary map file and the granularity ratio μ. As we can see, the storage cost decreases exponentially when μ increases. In particular, when $\mu = 10$, we only need 5000 bits for Middlesex county and 500 bit for District of Columbia to store the 4-tree / binary map file. This is much smaller than the size of the original TIGER/Line map. The retrieval of the VHC-map requires less than 0.1 seconds in our system, and the perturbation of a user's location requires less than 1 millisecond.

5.3. Evaluation of Anonymous Routing Component

We evaluated the communication QoS achieved by the anonymous routing component of CAP. Figure 13 depicts the cumulative distribution function (CDF) and probability density function (PDF) of time downloading the map image of 208,310 bytes from TIGER under the anonymous routing algorithms we proposed in Section 4. Diff/CA ($\leq$20KB/s) refers to differential routing with congestion avoidance whose tolerable throughput is 20KB/s. Table 1 gives the mean, median and confidence interval (95%) (CI) of the downloading time for different Tor routing algorithms.

We have a few observations from Figure 13 and Table 1: (i) The performance of Tor's default routing algorithm, weighted routing, can be intolerable for performance sensitive service such as LBS. The largest downloading time of the map image is 134.49s. (ii) The differential routing and the differential routing with congestion avoidance can significantly improve Tor's performance. With Diff/CA($\leq$20KB/s), the median downloading time is 5.23s compared with the weighted routing's 20.04s.

6. Related Work

Existing schemes on preserving location privacy in LBS can be generally classified into two categories: trusted third-party based and user based schemes.

Most research on trusted third-party based schemes adopts a k-anonymity based framework. In this framework, a trusted third-party called *anonymizer* is used to protect location privacy [9], [23]. For example, Gruteser *et al.* in [9] studied the k-area cloaking schemes in which the space is divided into a set of zones where each zone has at least k-sensitive areas.

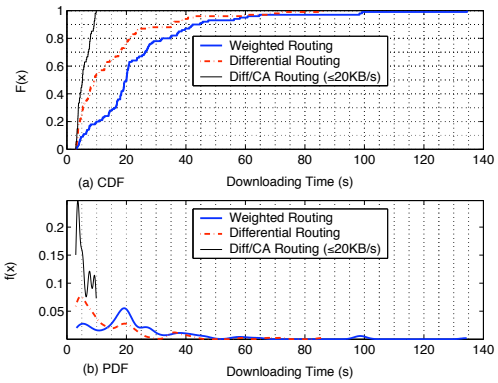

Figure 13. Download Time

Therefore, the adversary cannot identify which area the user visits. To relax the trusted third-party assumption, Mokbel *et al.* in [15] studied a scheme that leverages the peer-to-peer concept. However, the management of trust relationships among autonomous peers in LBS remains an open issue. A recent work removed the requirement of trusted third-party by using a private information retrieval (PIR) based scheme [7]. Most research on user-driven schemes adopts various obfuscation techniques at the user side aimed at protecting location privacy [2], [5]. For example, Duckham *el al.* in [5] studied the scheme to protect a user's real location by inserting some faked locations.

There has also been research on protecting location privacy by hiding users' network identities, such as network address. For example, Hu *et al.* in [10] presented a framework which uses random identity addresses such as IP and MAC addresses and adopts random silent periods in which mobile nodes don't transmit or receive frames. Not much work has been done on the QoS for anonymous communication networks. McCoy *et al.* in [14] plainly presented some results of Tor's performance measurement including router geopolitical distributions, circuit latency and throughput. Snader and Borisov [20] proposed to use bandwidth measurement algorithms and schemes that allow users to choose higher performance or higher anonymity.

7. Conclusion

In this paper, we developed CAP to address two challenging issues in privacy-preserving LBS: protection of user location privacy from both location data and network communication perspectives. CAP seamlessly integrates its location perturbation and anonymous routing components. We measure CAP in terms of location privacy, LBS query accuracy and communication QoS of the entire system. Its effectiveness is demonstrated by theoretical analysis, simulations, and experiments with an implemented prototype. Our work is the first end-to-end solution to protect location privacy and improve the accuracy of LBS while taking communication QoS into account. We believe that this paper lays the foundation for ongoing studies of privacy-preserving

Table 1. Downloading Time Comparison (unit: seconds)

	Weighted Routing	Diff	Diff/CA ($\leq$5KBs)	Diff/CA ($\leq$10KB/s)	Diff/CA ($\leq$20KB/s)
Median	20.0422	9.2733	8.9566	7.3709	5.2343
Mean	24.3192	15.0296	12.0749	8.6298	5.711
CI lower limit	19.9743	12.9606	9.8527	6.9204	5.1213
CI upper limit	30.0927	18.4387	14.6869	10.6894	6.4669

LBS.

Acknowledgement

This work was supported in part by the National Science Foundation under grants 0324988, 0329181, 0721571, 0808419, 0845644, 0852673, 0852674, and 0907964. Any opinions, findings, conclusions, and/or recommendations expressed in this material, either expressed or implied, are those of the authors and do not necessarily reflect the views of the sponsor listed above.

References

[1] B. Arai, G. Das, D. Gunopulos, and N. Koudas. Anytime measures for top-k algorithms. In *VLDB*, 2007.

[2] C. A. Ardagna, M. Cremonini, E. Damiani, S. D. C. di Vimercati, and P. Samarati. *Location privacy protection through obfuscation-based techniquess*. Data and Applications Security XXI (Lecture Notes in Computer Science), 2007.

[3] deluogps.com. Sirf star iii based mouse type USB GPS receiver for laptop, 2008.

[4] R. Dingledine and N. Mathewson. Tor: An anonymous internet communication system. http://archives.seul.org/or/talk/, 2006.

[5] M. Duckham and L. Kulik. A formal model of obfuscation and negotiation or location privacy. In *Proc. of the 3rd Internation Conference on Pervasive Computing and Communications*, 2005.

[6] geobytes.com. Ip address locator tool, 2008.

[7] G. Ghinita, P. Kalnis, A. Khoshgozaran, C. Shahabi, Tan, and Kian-Lee. Private queries in location based services: Anonymizers are not necessary. In *Proc. of ACM SIGMOD*, 2008.

[8] D. R. Glover and J. L. Simon. The effect of population density on infrastructure: The case of road building. *Economic Development and Cultural Change*, 23(3):453–468, 1975.

[9] M. Gruteser and X. Liu. Protecting privacy in continuous location-tracking applications. *IEEE Security and Privacy*, 2(2):28–34, 2004.

[10] Y.-C. Hu and H. J. Wang. Location privacy in wireless networks. In *Proc. of the ACM SIGCOMM Asia Workshop*, 2005.

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

[12] E. F. Krause. *Taxicab Geometry*. Dover, 1987.

[13] Y. Liu, Y. Gu, H. Zhang, W. Gong, and D. Towsley. Application level relay for high-bandwidth data transport. In *Proc. of the 1st International Workshop on Networks for Grid (GridNets)*, 2004.

[14] D. McCoy, K. Bauer, D. Grunwald, P. Tabriz, and D. Sicker. Shining light in dark places: A study of anonymous network usage. Technical report, University of Colorado at Boulder, 2007.

[15] M. F. Mokbel and C. Y. Chow. Challenges in preserving location privacy in peer-to-peer environments. In *Proc. of the International Workshop on Information Processing over Evolving Networks (WINPEN)*, 2006.

[16] Moonbuggy. Man accused of stalking with gps, 2004.

[17] H.-O. Peitgen and D. Saupe. *The Science of Fractal Images*. Springer-Verlag, New York, 1988.

[18] R. Pries, W. Yu, S. Graham, and X. Fu. On performance bottleneck of anonymous communication networks. In *Proc. of the 22nd IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, 2008.

[19] A. Research. GPS-enabled location-based services (lbs) subscribers will total 315 million in five years, 2006.

[20] R.Snader and N.Borisov. A tune-up for tor: Improving security and performance in the tor network. In *Proc. of the 15th Annual Network and Distributed System Security Symposium (NDSS)*, 2008.

[21] R. Sion and B. Carbunar. On the computational practicality of private information retrieval. In *Proc. of the 14th Annual Network and Distributed Security Symposium (NDSS)*, 2007.

[22] L. Sweeney. k-anonymity: a model for protecting privacy. *International Journal on Uncertainty, Fuzziness and Knowledge-based Systems*, 10(5):557–570, 2002.

[23] T. Xu and Y. Cai. Exploring historical location data for anonymity preservation in location-based services. In *Proc. of IEEE International Conference on Computer Communications (INFOCOM)*, 2008.

[24] N. Zhang and W. Zhao. Privacy-preserving data mining systems. *IEEE Computer*, 40(4):52–58, 2007.