A Privacy Enhancing Approach for Identity Inference Protection in Location-Based Services

Chowdhury S. Hasan, Sheikh I. Ahamed and Mohammad Tanviruzzaman
MSCS Department, Marquette University
Milwaukee, WI, USA
{chasan, iq, mtnviru}@mscs.mu.edu

Abstract—Recent advances in mobile handheld devices have facilitated the ubiquitous availability of location based services. Systems which provide location based services have always been vulnerable to numerous privacy threats. The more we aim at safe usage of location based services, the more we feel the necessity of a secure location privacy system. Most of the existing systems adopt the mechanism of satisfying k-anonymity which means that the exact user remains indistinguishable among k-1 other users. These systems usually propose the usage of a location anonymizer (LA) to achieve k-anonymity. In this paper we show that satisfying k-anonymity is not enough in preserving location privacy violation. Especially in an environment where a group of colluded service providers collaborate with each other, a user’s privacy can be compromised. We present a detailed analysis of such attack on privacy and propose a novel and powerful privacy definition called s-proximity. In addition to building a formal definition for s-proximity, we show that it is practical and it can be incorporated efficiently into existing systems to make them secure.

Keywords—Location privacy; k-anonymity; Location anonymizer

I. INTRODUCTION

The use of Location Based Services (LBS) has become ubiquitous with the growth of handheld devices, PDAs, smart phones and GPS enabled cars. LBSs will flourish even more with the surge of new genre of information systems. While requesting for a location based service, a user can easily mask his identity using unlinkable pseudonyms [1] but he needs to provide his location information, even if with less precision. The location service provider (LSP) or an adversary, who secretly listens to the communication channel between a user and the LSP, builds his own chronological record of location data over the period of time. Later this knowledge might be used by him with techniques like correlation attack, restricted space identification [2], observation identification [2, 3] etc to link the records with actual user identity. The user’s personal preference, state of health, political view etc can be even inferred from the places he visits or visited. No wonder, his mail boxes may be inundated with unwanted advertisements. This is location privacy violation and due to its possible aftermaths preservation of an individual’s location privacy is of utmost importance.

The privacy of the query issuer in a location based query is preserved by hiding her exact location with a Cloaked Region (CR) [4, 5, 6] or Anonymous Spatial Region (ASR) [7] forwarded to the service provider. Along with the query issuer inside the CR, there exist k-1 other users who has previously issued or exists with potential to issue a query. The task of Location Anonymizer (LA), a trusted party is to generate the appropriate CR that thwarts any attempts on exact location inference by the attacker through the unsecured channel between the Anonymizer and the Service Provider. The k-anonymity [8] principle adopted in all the approaches [4-7] ensures that the CR chosen for a query offers the attacker a probability of re-identification not exceeding 1/k, k being the preferred anonymity level of query issuer.

But in this paper we argue that k-anonymity does not provide sufficient protection against privacy violation. We present two attacks, the heterogeneity attack and the conformity attack, and show how they can be used to compromise a k-anonymous location based query. The heterogeneity attack reveals that k-anonymity can create groups that fail to provide overall anonymity due to lack of sufficient match among the members with respect to some sensitive user attribute. Likewise, approaches satisfying only k-anonymity disregard consequences of revealing important context [9] information though the service request and pave the way for conformity attack. Besides illustrating the attacks with real world examples, we have provided their formal definitions which clarify how they relate to the contexts [9] of the query and static information [10] of the users in the anonymity set (AS) [10]. Most of the existing approaches [3-7, 14, 16-18] are vulnerable to these attacks as they undergo following problems.

1) They choose k number of users for constructing a CR on the basis of current locations of the users only ignoring any other relevant static attribute of the users being grouped together.
2) Most of the approaches forward the query to the LSP without making any modification. However, a query that contains request for a specialized service may disclose a number of contexts and static user information on those contexts hastens re-identification.
3) They do not consider preference of users regarding any other contexts of their interest beyond location on the basis of which they want anonymization.
As an attempt to guard against the above mentioned attacks we have introduced a new notion of privacy, called \textit{s-proximity}, which requires that each anonymity set (AS) contains at least \( s \) members belonging to the equivalence class of the query issuer. An equivalence class is defined to consist of users having high correlation with the actual query requester with respect to a set of static user attributes [10]. With this new privacy parameter a user’s privacy profile [6] takes the form of \(< k, s, A_{\text{min}} >\). In this paper we propose a pragmatic solution that offers service with such privacy protection. Our approach uses a trusted third party to mediate user’s query to the LSP. We call this trusted third party Context Aware Location Anonymizer (c-LA) as it is featured with additional modules for context based query generalization, proximity group formation as pre-steps of CR generation. Our approach uses a novel algorithm called Selective Nearest Neighbor (SNN) for AS construction and CR generation. A formal proof establishes that SNN provides enhanced privacy by reducing the probability of re-identifying the actual query issuer. Implementation of the system validates the feasibility of our proposed approach.

The rest of the paper is organized as follows. Section II depicts with examples how \( k \)-anonymity can come under attacks. Section III walks through the definition and notations used in the formalization of the attack model. We present the definition of \( s \)-proximity in Section IV. We discuss our approach in detail in Section V. Section VI focuses on the related works. Evaluation results are presented in Section VII. Section VIII concludes our findings and provides avenues for future research.

II. MOTIVATION

Let’s take a look at some real world examples of location based services to better understand the threat of privacy violation. The examples demonstrate that existing solutions which depend on satisfying \( k \)-anonymity are still vulnerable to privacy violation attacks.

We assume Alice is currently subscribed to a location based system which uses a trusted LA to make her location \( k \)-anonymous and forward her query to the LSP. Alice is guaranteed that her exact location is never disclosed to the LSP by assuring that she remains \( k \)-anonymous to the unknown third-party where \( k \) is chosen by herself according to her required privacy level. However, we present couple of random scenarios where her ultimate privacy is shown to be endangered although her location is \( k \)-anonymous to the LSP.

Scenario 1: Alice, owing to some chronic disease, goes through regular medical checkup. She moves to a new place and looks for the nearest medical center. Due to the specific nature of her illness she is considering only healthcare centers that treat feminine diseases. This makes it logical for her to query for the nearest female hospital from her current location. She submits her query regarding locating the nearest female hospital to the LSP. The LBS system that handles her request provides her with the requested service along with preserving her location privacy by means of ensuring that her location information is made \( k \)-anonymous before being submitted to any un-trusted party (in this case, LSP).

Scenario 2: Here Alice uses the LBS system for her academic purpose. As she starts her new semester in the graduate school, after the first week of classes she is given a list of books some of which she needs to purchase urgently. She searches for the nearest bookstores from her residence and is served by the LBS system. The LBS system takes her location and returns her the list of book stores located nearest to her place.

Scenario 3: Alice makes frequent travels to new places where she faces the problem of finding nearest car parks. She is reluctant to provide her exact location, rather the LA makes sure that instead of her location a CR that contains at least 3 other users is sent to the LSP. Thus the LBS system tries to protect her location privacy and finally preserve any un-solicited identity disclosure. Still her identity can be disclosed to the LSP as discussed below.

A. Attacks on \( k \)-Anonymity:

We assume that the LSP or any other adversary has access to the following information [7].
1) Accesses requested anonymity level (i.e. value of \( k \)) for each query.
2) Recognizes the users belonging to the AS corresponding to the CR of a query.
3) Collects static profile data of all the users that have made query at some time.
4) Identifies any special context revealed by the query and maps that context to relevant user attribute.

Finally it is supposed that a group of malicious LSPs collaborate with each other by sharing their knowledge about the users in an effort to re-identify any individual query requester.

Based on the above assumptions we summarize the knowledge of the LSPs involved in the LBS system in the following table.
The service being requested in the context of the service is a query for a nearest female hospital. We consider that the set members of the service consists of location-based services only. To better understand the notion of this novel attack we define it formally in the following section.

III. ATTACK MODEL

Now we move into the formalization of the attacks. However, prior to that a walk through the notions and definitions will be worthwhile.

A. Definitions

User Set (U): The LA maintains a list of users who have subscribed to it. This list is denoted by \( U = \{u_1, u_2, ..., u_N\} \) where \( u_i \) represents \( i^{th} \) user.

Context (C): Any sensed information used to describe some physical phenomena is defined as context [10]. In this paper we mean by context any deterministic condition or situation that characterizes a service. The contexts can take on different levels of granular values. A finite domain of information for all the contexts in the application is assumed in our proposed model. Thus, the individual sets of contexts will have a finite number of possible values. Using higher level granular values the service becomes more generalized.

Service (S): We consider, the set \( S = \{s_1, s_2, ..., s_m\} \) consists of location-based services only. A user has to provide some sort of location data to avail such services. Although location is the major emphasized context for these services, other contexts are also associated with each specific. Instead the request for a nearest hospital involves fewer contexts [9, 10].

**Observation:** The query is made more generalized by changing granularity level [10] of the intended service.

Both of these two types of attacks have some common properties as they relate to the context of the query and the static attribute of the requester. To better understand the notion of this novel attack we define it formally in the following section.

<table>
<thead>
<tr>
<th>Criteria</th>
<th>Scenario 1</th>
<th>Scenario 2</th>
<th>Scenario 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Query</td>
<td>Nearest Female Hospital</td>
<td>Nearest Book Store</td>
<td>Nearest Car Park</td>
</tr>
<tr>
<td>LSP</td>
<td>LSP (_i)</td>
<td>LSP (_j)</td>
<td>LSP (_k)</td>
</tr>
<tr>
<td></td>
<td>AS (value of (k))</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>Members of AS (LSP has</td>
<td>{Alice, Bob, William, Ada}</td>
<td>{Alice, Carl, Jacob, Michael}</td>
</tr>
<tr>
<td></td>
<td>identified)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Context of the service</td>
<td>Healthcare</td>
<td>Academic</td>
<td>Transportation</td>
</tr>
<tr>
<td>Relevant user attribute</td>
<td>Gender</td>
<td>Occupation</td>
<td>Driving License</td>
</tr>
<tr>
<td>Findings</td>
<td>Only Alice is female in the group</td>
<td>Only Alice is student in the group</td>
<td>Only {Alice, Joshua} have Driving License</td>
</tr>
<tr>
<td>Identity Inference (probable query requester)</td>
<td>Alice</td>
<td>Alice</td>
<td>{Alice, Joshua}</td>
</tr>
</tbody>
</table>

In Scenario 1 the attacker (in this case LSP \(_i\) ) successfully identifies Alice to be the query issuer. The fact that she was grouped with 3 male users who seem unlikely to request for a female hospital enabled the re-identification. Her query was too specific as well.

In Scenario 2 the attacker succeeds again. Here Alice is grouped with non academic users which made the attacker guess Alice to be the query requester.

In Scenario 3 the attacker was able to narrow down the list of possible issuers. From the context of the service being requested the attacker knows that any user not having driving license is unlikely to request for a car park.

There might be numerous occasions such as discussed above where \( k\)-anonymity fails to guard against identity disclosure. We have classified all these into two categories which are discussed below.

B. Classification of Attacks

1) **Heterogeneity Attack:** The members in the anonymity set are too much diversified with respect to some static attributes. In worst case, the query requester might possess some exclusive identifiable property, then she no longer remains indistinguishable. For example, an AS which groups a female user with all male users is vulnerable to such attack.

**Observation:** The anonymity set should ensure the inclusion of a minimum number of users with similar profile as the actual query requester with respect to some static attributes.

2) **Conformity Attack:** The service being requested in the query is too specialized as it relates to some particular contexts. A user has to conform to some specific conditions to be a potential candidate for such service. If most of the users in the AS fail to possess those properties, chances of the actual requester being re-identified increase. For example, a query for a nearest female hospital is too specific. Instead the request for a nearest hospital involves fewer contexts [9, 10].

**Observation:** The query is made more generalized by changing granularity level [10] of the intended service.
service. Generally each service is provided by an LSP whereas it is common for an LSP to deliver a group of services.

**Static Information (SI):** The static information could be the user’s static attributes or credentials used to authenticate his identity. We refer to the set of static information as $SI = \{s_{i1}, s_{i2}, ..., s_{ik}\}$. We use the terms static information, static user attribute and static user profile interchangeably throughout the paper. In some places we have used $SI = \{a_1, a_2, a_3, ..., a_k\}$ where $a_i$ stands for $i$th static user attribute. Values of these attributes specify individual query requesters. A subset of $SI$ forms quasi identifier [11]. The static information is not directly provided by the requester of a location based query rather an attacker collects it from background data sources.

**Anonymity Set (AS):** The list of probable issuers of a query request is called an anonymity set. The request can be issued for a location based query rather than attacker collects it from background data sources.

Now, we suppose the following:

$s_j$, $v(c_i)$: Value of the context $c_i$ of service $s_j$

$u_r$, $v(s_l(c_l))$: Value of the static user attribute corresponding to context $c_l$ of user $u_r$. Based on the above assumptions we formally define the attack model below.

Suppose, $u_r$ submits query $r = (s, c, si)$ to the LSP with cloaked region formed by $AS = \{u_1, u_2, ..., u_k\}$ to meet her k-anonymity requirement.

We assume that, $u_r$ is distinctive from every other user in $AS$ with respect to any static user attribute corresponding to some context of the service being requested, i.e., $\exists c_l \in c[u_r, v(s_l(c_l))] \neq u_r, v(s_l(c_l)), \forall j \neq r$ (1)

Now, the LSP being a malicious attacker finds out some context $c_a$ relevant to query $r$ and $si_a$, be a static user attribute corresponding to $c_a$. Then it looks up in background data sources to collect data of $si_a$ attribute for all $u_l \in AS$.

Since $u_r$ is the actual query requester, $u_r$, $v(s_l(c_a)) = s, v(c_a)$ ... (2) holds true.

According to Assumption (1):

$\exists j|u_r, v(s_l(c_a)) \neq u_j, v(s_l(c_a)), j \neq r$ ...(3).

The attacker can exclude from the $AS$ the users that satisfy eqn. (3). As the size of $AS$ shrinks, the probability of re-identifying $u_r$ increases. Collaboration among a group of malicious LSPs may yield a context $c_x$ for which eqn. (2) may hold for every $u_l \in AS$ except $u_r$. Then $u_r$ could be immediately re-identified as the query issuer.

As we have already defined the identity inference attack in a formal way, we proceed to introduce our approach to handle it. At the core of our solution there exists a new privacy parameter called $s$-proximity which we introduce next.

IV. $s$-PROXIMITY: A NEW PRIVACY PARAMETER

A close look at the attack scenarios reveals that if actual query requester is fully distinguishable from other users in the AS, with respect to some static attribute relevant to the context of the query, her identity may be disclosed immediately whatever her achieved location anonymity may be. Therefore, it is highly desirable that at least a minimum number of users in the AS have similar profile as the actual query issuer. This requirement adds a novel parameter, called $s$-proximity, to the users’ privacy profile [21]. Before defining $s$-proximity, we need to introduce couple of relevant definitions.

**Dissimilarity Measure:** This metric measures the amount of divergence between two users with respect to a certain static user attribute. We use the notation $d_{stp}(u_m, u_n)$ to denote the dissimilarity measure between $u_m$ and $u_n$ based on $si_p$ where $si_p$ is a static attribute corresponding to the context $c_p$ of service $s_j$. $d_{stp}(u_m, u_n) < \delta \Rightarrow u_m \sim u_n$ ($u_m$ is “similar to” $u_n$), $\delta$ is a user defined threshold value.

**Equivalence Class (E):** The set of users that are similar to $u_i$ with respect to $p$th static attribute $si_p$ is called equivalence class of $u_i$ and denoted by $E_{si_p}(u_i)$.

$E_{si_p}(u_i) = \{u_j \in U | (u_i \sim u_j) \wedge (si = si_p)\}$

$= \{u_j \in U | d_{stp}(u_i, u_j) < \delta\}$

**Divergence Class (D):** The set of users that are not similar to $u_i$ with respect to $p$th static attribute $si_p$ is called divergence class of $u_i$ and denoted by $D_{si_p}(u_i)$.

$D_{si_p}(u_i) = \{u_j \in U | (u_i \sim u_j) \wedge (si \neq si_p)\}$

$= \{u_j \in U | d_{stp}(u_i, u_j) \geq \delta\}$

$s$-Proximity: By $s$-proximity we mean that the AS will contain at least (s-1) other users similar to the actual query requester $u_r$, i.e., $|AS \cap E_{si_p}(u_r)| \geq s$. 

Authorized licensed use limited to: Marquette University. Downloaded on July 13, 2010 at 05:18:41 UTC from IEEE Xplore. Restrictions apply.
Selecting higher value of $s$ guarantees strong privacy but at the cost of degraded quality of service. So, users themselves are responsible for choosing value of $s$ according to their preference.

**Enhanced Privacy Profile ($k,s,A_{min}$):** With the introduction of $s$-proximity our model assumes that a user’s privacy profile consists of $k,s$ and $A_{min}$ which stand for anonymity requirement, proximity requirement and minimum CR area constraint.

V. OUR APPROACH

We start our discussion with the generalized view of a location privacy framework having a trusted location anonymizer (LA). In such systems subscribed users send their location based query to the LA which replaces the exact location with a cloaked region and forwards the query to the LSP. In reply LSP returns the list of query results which is usually termed as list of point of interests (POI) to the LA and eventually the POI list is forwarded to the query requester. The system is depicted in the figure below.

![Figure 1. The Location query processing through LA](image)

Such a system tries to preserve user’s location privacy by implementing the measure of $k$-anonymity. As long as a system ensures that user’s location is $k$-anonymous to the LSP, it apparently succeeds in preserving location privacy. In this paper we have shown, this typical notion of safeguarding location privacy by means of $k$-anonymity is not adequate rather it may endanger a user’s ultimate privacy by revealing her sensitive private data. To provide a robust privacy solution we need to ensure both $k$-anonymity and $s$-proximity. We use this existing privacy framework and incorporate advanced functionalities into the LA to provide such a solution.

A. Overview of Our Approach

We propose a location privacy system with a trusted LA which creates anonymization group considering context of the query. At the core of our approach lies an enhanced location anonymizer attributed with multiple capabilities and we call it a Context Aware Location Anonymizer (c-LA). As the basic functionality is same we use the terms LA and c-LA interchangeably hereinafter. The main focus of the solution is on minimizing the probability of re-identifying the actual query requester along with anonymizing his location information. A location based query usually contains other sensitive information alongside spatial data. Hence, it is not enough to hide only the location data rather we propose modifying the query as a whole to minimize any identifying information it carries. We term this process **Query Generalization** which is the first step of our solution. The task is accomplished at the LA by an additional module called **Query Analyzer** which identifies any sensitive context in the query and looks for possible generalization. Only generalizing the query does not solve all the problems. As we discussed earlier the way users are grouped together to form anonymization set impacts the possibility of re-identifying actual query issuer. The task of satisfying the $s$-proximity condition is performed by the module called **Partitioning Agent** responsible for splitting the entire user set into **Equivalence Class** and **Divergence Class**. Finally, the **CR Construction Unit** generates the cloaked region based on anonymization set created from users in the **Equivalence Class**.

![Figure 2. Architecture of Context Aware Location Anonymizer (c-LA)](image)

B. Details of Our Solution

The Context Aware Location Anonymizer, as depicted in Figure 2, consists of multiple units, each unit performing dedicated tasks. The overall process of anonymization is accomplished in several steps which are discussed below in detail.

**Step 1. Initialization:** Location Anonymizer maintains a list of services for which it has registered with corresponding LSPs. For each service, LA has knowledge of relevant contexts using which it generates a set of static information interrelated with the service. LA stores all these information in the **Service Attribute Mapping (SAM)** table which has the form:

```xml
< Service,Context_List,Static_Information_List >
```

The LA informs its subscribed users about its SAM table. A user chooses the services of his interest and finds the set...
of required static information. The user then registers with the LA providing that static information.

![User Profile](image)

**Step 1:** send SAM table

**Step 2:** send <serv_pref_list, per_prof, priv_prof>

At the time of registration the user also notifies the LA about his privacy profile. The user does that in Step 2 in the above figure by passing the values of <k,s,Amin > denoted by priv_prof. In the same message the user includes his personal information in per_prof and his preferred services list in serv_pref_list. At the end of this phase the LA gets necessary information from the subscribed users to fill in the User Profile table. Structure of this table is:

< User, serv_pref_list, per_prof, priv_prof >

**Step 2. Query Generalization:** In this step the LA tries to generalize the query by modifying the content of the request. In effect, it modifies the granularity level of the requested service. To facilitate that all the services are arranged in a hierarchical tree structure where a node represents the generalized service for all the services in the sub-tree rooted at that node. The contexts related to the services are used to determine the correlations among them and to construct the tree. This step may be termed as the Service Generalization step as well. The main task of service generalization is accomplished by using a generalization function. In order to better understand how the generalization function works we intend to define the query request in a formal way as follows.

A query request consists of the identity of the user, the context and the static information and it has a complex domain: \( R \in \{ U \times \Pi_1, C_i \times \Pi_1, S_j \} \). The context and static information contained in this request are modified by the generalization function, \( G_s \) to yield a generalized request. The generalized request domain can be represented as: \( R' \in \{ U' \times \Pi'_1, C'_i \times \Pi'_1, S'_j \} \) which contains the context and static information augmented to higher granularity levels. We have identified the following properties [9] which the generalization function \( G_s : R \rightarrow R' \) holds.

1. **Many to one mapping:** Two or more requests can be transformed into same augmented request that is forwarded to the service provider. \( \exists r_1, r_2 \in R, G_s(r_1) = G_s(r_2) = r', r_1 \neq r_2 \)

2. **Idempotent with generalized request:** If the generalization function is applied to a generalized request, no more generalization will be possible, provided the generalization criteria along with anonymity and proximity levels remain the same. \( \forall r \in R, G_s^n(r) = G_s(r), n = 1, 2, ... \)

3. **Invertible:** The generalization function also has an inverse function. \( \forall r \in R, \exists r_1, r_2, G_s^{-1}(r) = r_1, G_s^{-1}(r_2) = r_2 \)

4. **Asymmetric:** The generalization function and its inverse are asymmetric in nature. \( \forall r \in R : G_s^{-1}(G_s(r)) \neq r, \) where \( G_s^{-1} : R' \rightarrow R, n \geq 1 \)

5. **Non injective:** The generalization function is non injective in nature. If two or more requests are generalized using the same augmented request it doesn’t imply that the requests are the same. \( \forall r_1, r_2 \in R : G_s(r_1) = G_s(r_2) \neq r_1 = r_2 \)

6. **Non Equivalence of function:** The generalization functions having different privacy preferences may provide the same generalization for two or more requests. \( \forall r \in R : G_s^1(r) = G_s^2(r) \neq G_s^1 \equiv G_s^2 \). Although some of the requests achieve same generalizations, \( G \) may not be equivalent due to the fact that all of the context or static values are not assigned, or the granularity level is coarse enough so that the generalization was not applied even.

**Step 3. Proximity Group Formation:** The user submits his location based query along with her privacy preference parameters \( k, s, A_{min} \) to the LA. In our proposed model user has option to send a set of additional parameters which set the priorities of the static information variables involved with the service being requested. Based on these inputs the LA partitions the entire user set into two disjoint subsets: Equivalence Class and Divergence Class according to the following algorithm.

**Algorithm 1: Proximity Group Formation**

Input: query \( r \in R : < s, c, s_i > \), weight matrix \( w \)

1. Sort si in descending order of \( w \)

2. for \( (i = 1 to k) \) do

3. for \( (i = 1 to N) \) do

4. if \( (d(U[j], si[i], U[r], si[i]) \leq \delta) \) then

5. for \( (j = 1 to N) \) do

6. if \( (U[j] \notin E) \) then

7. return \( E, D \)

The algorithm takes the query, \( r \) and weight matrix, \( w \) as input. \( w \) contains user’s preferred priority of the involved static attributes. Based on that priority other users are compared with the query requester. The users that are similar to query requester are inserted into equivalence class, \( E \) and others are inserted into divergence class, \( D \).

**Step 4. Cloaked Region Generation:** In this step the LA constructs the cloaked region which is forwarded to the LSP. First, it chooses the AS in such a way so that it meets both \( k \)-anonymity and \( s \)-proximity. Then using the locations
of the members of AS the CR is constructed which meets the \(A_{min}\) requirement. The CR generation process follows the algorithm presented below.

\textbf{Algorithm 2: Selective Nearest Neighbor (SNN) Cloak}

\textbf{Input:} \(U, E, k, s, A_{min}, L\)

\textbf{Initialization:} \(AS = \{\}, i = 0\)

1. Sort users in \(U\) in ascending order of dissimilarity from \(U[r]\)
2. while \((|AS| < k)\) do
3. \hspace{1em} \textbf{if} \((|AS| < s)\) then
4. \hspace{2em} \textbf{if} \((U[i] \in E)\) then
5. \hspace{3em} Insert \(U[i]\) into \(AS\)
6. \hspace{2em} \textbf{else} \(i++\)
7. \hspace{1em} \textbf{else} Insert \(U[i]\) into \(AS\)
8. \textbf{call} GenerateCR\((L, AS)\)
9. return \(CR\)

The equivalence class of the query issuer, \(E\), privacy profile \((k, s, A_{min})\) and location vector, \(L\) are supplied to the algorithm as input. It first tries to meet \(s\)-proximity by inserting \(s\) nearest users into AS taken from \(E\). Then it inserts other \((k-s)\) nearest users into the AS to meet \(k\)-anonymity. Finally the AS along with location vector, \(L\) is used to construct the CR.

\textbf{C. Attack Prevention: Formal Proof}

We conclude this section by showing that our CR generation algorithm SNN-Cloak is attack resistant.

\textbf{Lemma:} \(c - LA\) reduces the probability of re-identification of actual query issuer.

\textbf{Proof:} Let, \(u_r \in U\) submits query \(r = (s, c, si)\) to the LA with her \(k\)-anonymity and \(s\)-proximity requirement. The LA constructs two different anonymity sets \(AS_1\) and \(AS_2\) applying SNN algorithm and NN algorithm [7] respectively. \(AS_1\) meets both \(k\)-anonymity and \(s\)-proximity requirement however \(AS_2\) meets only \(k\)-anonymity.

Suppose, \(c_x\) be a context relevant to query \(r\) and \(si_x\) be a static user attribute corresponding to \(c_x\). Since \(u_r\) is the actual query requester, \(u_r, v(si[c_x]) = s, v(c_x)\) ...(1)

Let, \(E_x\) be the equivalence class of \(u_r\) with respect to \(si_x\). Then, \(u_r, v(si[c_x]) = s, \forall u_j \in E_x\) ...(2).

We define, \(E_{SNN} = E_x \cap AS_1\) and \(E_{NN} = E_x \cap AS_2\). Hence, \(u_j, v(si[c_x]) = s, \forall u_j \in AS_1\) ...(3) [ SNN meets \(s\)-proximity] Conversely, \(\exists u_j \in AS_2\) \(u_j, v(si[c_x]) \neq s, \forall u_j \in AS_1\) ...(4) [ NN does not meet \(s\)-proximity] Using (3) and (4) we get \(|E_{SNN}| > |E_{NN}|\) ...(5).

In \(AS_1\) the probability of re-identifying \(u_r\) based on \(c_x\) and \(si_x = \frac{1}{|E_{SNN}|}\)

Similarly, In \(AS_2\) the probability of re-identifying \(u_r\) based on \(c_x\) and \(si_x = \frac{1}{|E_{NN}|}\)

From (5) \(\frac{1}{|E_{SNN}|} < \frac{1}{|E_{NN}|}\). So, the SNN algorithm applied by \(c - LA\) reduces the probability of re-identification of actual query issuer.

\textbf{VI. RELATED WORKS}

Ours is an approach to provide location privacy solution satisfying \(k\)-anonymity along with \(s\)-proximity using a location anonymizer. A thorough survey of literature reveals that lots of works have been done to deal with location privacy but none has proposed the inclusion of parameter \(s\)-proximity. Existing approaches in achieving anonymity for the LBS services [3, 5, 12, 13, 14] have ignored the fact that static information is required during the service access. In [15] it is shown that the knowledge of the attacker can be used to perform the re-identification attack. The selection of quasi identifiers [8] from contextual information can place the individual privacy at serious risk [10].

Most of the location-privacy preservation frameworks use Location Anonymizer (LA) to meet the requester’s \(k\)-anonymity requirement by providing a spatial region namely the Cloaked Region (CR) where the requester is indistinguishable from \(k\)-other request issuers. Gruteser and Grunwald were the first to introduce such cloaking [2]. They calculated CR based on Quad-tree where they recursively partition the space into four equal squared regions until the users fit in a quadrant where their privacy requirement is satisfied. Another quad-tree variant using pyramid structure called New Casper by Mokbel et al [6] achieves superior worst-case complexity for calculating CR over the interval cloaking algorithm [4]. In order to optimize CR region Gedik et al. proposed Clique-Cloak [16] and Mokbel et. al. used Nearest Neighbor cloak [7]. Hilbert Cloak [14] uses the Hilbert space filling curve to map the 2-D space into sorted 1-D values which are partitioned into groups of K users. All of these approaches try to optimize the CR construction technique to minimize CR size. While constructing the CR, they consider only location context disregarding any other context of the query. Due to this, the CR generated in these approaches yield anonymity sets which are vulnerable to the heterogeneity attack and conformity attack proposed in this paper.

There are other categories of location privacy frameworks which do not use any LA. Matt Duckham et al [17] used an obfuscation technique to protect location privacy without LA. Techniques described in [18, 19] eliminated anonymizer by considering mutual trust among the entities. The technique by Ghinita et. al. [18] uses a variant of Private Information Retrieval (PIR) theory known as Computational PIR (CPIR) for finding the approximate and exact nearest neighbors of the Point of interests (POI). These approaches also fail to protect identity inference caused by the attacks we have shown.
In [11, 20] it is shown how individual’s identity can be inferred even from a k-anonymized data set. They proved that k-anonymity is not a sufficient measure against re-identification attacks. To protect this, the notion of l-diversity is proposed in [11] whereas [20] shows the weakness of l-diversity and eliminates those by introducing the concept of t-closeness. These works are similar to our approach. They showed the identity inference attacks in scenario of micro-data publishing whereas we have formulated the attacks in case of using location based services. The generic view of the problem was addressed while considering the disclosure of a number of contexts and static information involved during the service access [10]. They presented the concept of contexts of a query and related static user information from a theoretical viewpoint. We provide a practical solution of the problem.

VII. EVALUATION

We have implemented a prototype version of our proposed system. The modules of context aware location anonymizer were developed on a machine with hardware configuration Intel Processor 1.7 Ghz, 1.5 GB Memory and Windows Vista as OS. We have deployed an application that uses, on client side, a Dell Axim X50v pocket PC (Processor type is Intel 624 MHz Xscale, ROM is 128MB Flash). The underlying OS is WinCE and the implementation language is C# on .NET Compact framework.

The spatial data used in the evaluation were taken from North American data set [1] consisted of 15K points which were used by clients and the LA as user points in 2D space. Figure 4 depicts how the client module works. The initialization step consisting of the user registration tasks are shown in Figure 4 (a) (b). The next three figures display how a registered user submits location based query to the LA and gets reply subsequently.

We have evaluated performance of our system and the findings are summarized in Figure 5. Performance of the system was measured in terms of the metrics: Query Success Rate, CR Construction Time and CR Size (absolute/relative). The percentage of time a user was provided with her required service was denoted by Query Success Rate. Other two metrics measured the time required for constructing a cloaked region and the size of the constructed CR. As Figure 5 (a) depicts, we achieved overall high success rate though it reduced a bit with higher proximity requirement. The graphs in Figure 5 (b) (c) demonstrate that the construction time and size of a cloaked region increases with higher proximity level. These parameters also show higher values for increased number of subscribers. Performance of our prototype implementation was compared with couple of other existing systems (NNC [7], IC [4], Casper [6], HC [14]) and it is found that our approach yields cloaked region with a bit large size (shown in Figure 5 (d)) which is quite acceptable considering the enhanced level of privacy it offers compared to the existing frameworks.

From the experimental facts it is evident that our proposed framework with a context aware location anonymizer is really feasible to be implemented in real world LBS system. Performance of the system is acceptable as compared to existing systems. The system implements both s-proximity and k-anonymity which is a novel approach capable of safeguarding the attacks presented in this paper.
Figure 5. Evaluation Results (a) Query Success Rate (b) Variation of CR Construction Time according to requested proximity level (c) Variation of CR Construction Time according to requested anonymity level (d) Comparison of our approach with existing approaches in terms of relative CR size.

VIII. CONCLUSION

In this paper we have proposed a location privacy solution with a trusted third party (c-LA) equipped with additional functionalities which are accomplished prior to AS construction. The algorithms presented in the paper aim to construct an AS which meets k-anonymity and s-proximity. The novel privacy feature s-proximity is proposed as a solution of couple of plausible attacks applicable against most of the existing approaches. We have implemented a prototype version of the system. Evaluation results demonstrate the feasibility of our proposed approach along with its performance measures.

REFERENCES


