Extended Privacy in Crowdsourced Location-Based Services Using Mobile Cloud Computing

Jacques Bou Abdo, Thomas Bourgeau, Jacques Demerjian, Hakima Chaouchi

Hindawi Publishing Corporation
Mobile Information Systems
Volume 2016, Article ID 7867206, 13 pages
http://dx.doi.org/10.1155/2016/7867206

Research Article

Extended Privacy in Crowdsourced Location-Based Services Using Mobile Cloud Computing

Jacques Bou Abdo,[1] Thomas Bourgeau,[2] Jacques Demerjian,[3] and Hakima Chaouchi[4]

[1] *Computer Science Department, Notre Dame University-Louaize, Zouk Mosbeh, P.O. Box 72, Lebanon*
[2] *UPMC, Sorbonne University, LIP6, Paris, France*
[3] *Faculty of Sciences, Lebanese University, LARIFA-EDST, Pierre Gemayel Campus, Fanar, Lebanon*
[4] *Telecom SudParis, Institut Telecom, CNRS SAMOVAR, UMR 5751, 9 rue Charles Fourier, 91011 Evry, France*

Correspondence should be addressed to Jacques Bou Abdo; jbouabdo@ndu.edu.lb

Received 28 January 2016; Accepted 19 June 2016

Academic Editor: Michele Amoretti

Crowdsourcing mobile applications are of increasing importance due to their suitability in providing personalized and better matching replies. The competitive edge of crowdsourcing is twofold; the requestors can achieve better and/or cheaper responses while the crowd contributors can achieve extra money by utilizing their free time or resources. Crowdsourcing location-based services inherit the querying mechanism from their legacy predecessors and this is where the threat lies. In this paper, we are going to show that none of the advanced privacy notions found in the literature except for K-anonymity is suitable for crowdsourced location-based services. In addition, we are going to prove mathematically, using an attack we developed, that K-anonymity does not satisfy the privacy level needed by such services. To respond to this emerging threat, we will propose a new concept, totally different from existing resource consuming privacy notions, to handle user privacy using Mobile Cloud Computing.

1. Introduction

Mobile Cloud Computing (MCC) is a very promising technology supported by major Cloud Service Providers (CSPs), mobile operators, and mobile vendors. CSPs are even offering Cloud platforms (such as Google Cloud Platform [1]) to be utilized by 3rd-party development companies, which will result in a boom in mobile Cloud applications. Some of the applications might have similar features forcing platform developers to thrive for a competitive edge to make their products profitable. In this highly competitive industry, an application's survival is critically based on its capability to respond to users' preferences. Mobile Cloud tries to satisfy user's requirements by offering extensive computation/storage resources which are accessible through mobile networks and decrease the mobile's power consumption. However, these resources are not always enough to meet customer needs.

Although computers' intelligence and processing power are drastically increasing, search engines are still bounded to factual answers and fail in providing personal opinions which are more valuable [2] and could only be provided by human interaction.

Crowdsourcing is broadcasting tasks, used to be executed by machines or employees, into an external set of people (contributors) [3, 4], as shown in Figure 1. Each of the crowd contributors has a pool of expertise that spans across different categories such as topic, language, geographic location, age, gender, and education.

When a request (outsourced task) is sent, the crowdsourcing server evaluates it and retrieves the expertise needed for successful execution. The contributors matching the needed expertise are then selected to respond to the request.

Crowdsourcing can be implemented in different applications and scenarios such as the following:

(i) "*Social search engine*" [4] is one of the most popular crowdsourcing application themes which focuses on answering context-related questions using human help (crowd) instead of or complementary with search

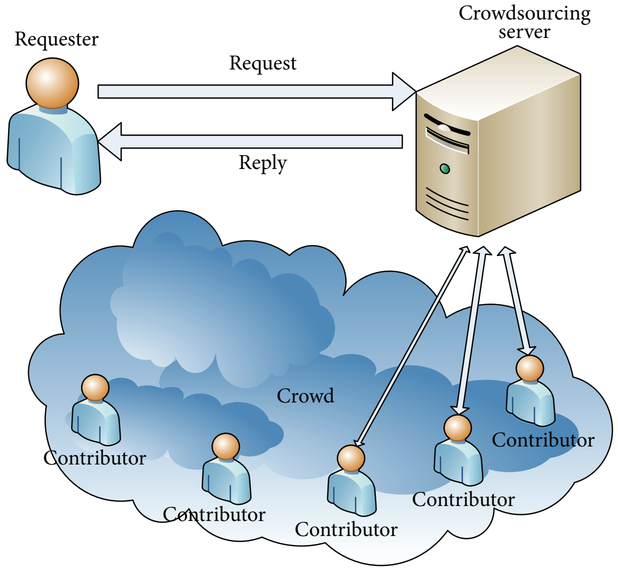


Figure 1: Crowdsourcing scenario.

engines [5–7]. Chacha [8] is a popular crowdsourcing application.

(ii) "*Crowdsourced location-based service*" [4] is a method to fetch the recommendations about certain location-based categories posted by people with taste and interest similar to the requester. Foursquare [9] is a popular application offering crowdsourced location-based service (LBS).

In this paper, we will be focusing on "crowdsourced location-based service" (crowdsourced LBS), but our work could be extended to other types of applications. Chorus [10], Foursquare [9, 11], and CrowdSearcher [2] are three examples of crowdsourced LBS applications which require user's location to be transmitted with the request in order to ensure optimized task routing.

Figure 2 shows a sample request sent to CrowdSearcher and the steps followed in the resolution mechanism. The request is as follows: find a good job with a suitable apartment close to a good drug rehabilitation center within my neighborhood (i.e., near my current location). The steps taken to respond to the above query are as follows:

(1) The resolution mechanism searches for jobs suitable for the requestor's profile and houses offered for rent, both located in the requestor's neighborhood.

(2) The retrieved jobs and their offered salaries are compared based on the crowd's opinion. The selected crowd has to have expertise in the requester's job domain and geographic location.

(3) The drug rehabilitation centers close to the retrieved houses are compared based on the crowd's opinion. The second crowd has to have expertise in health or social companionship and the studied geographic location.

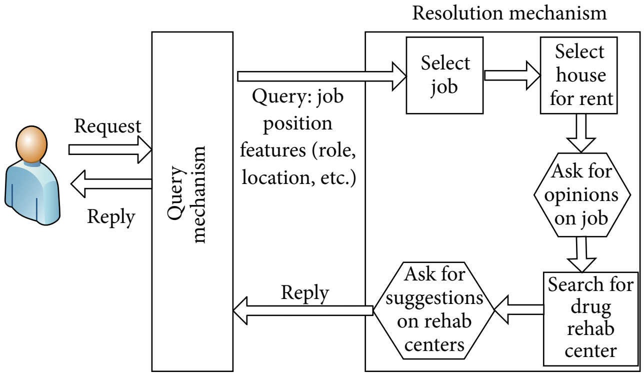

Resolution mechanism
Request
Reply
Query mechanism
Query: job position features (role, location, etc.)
Select job
Select house for rent
Ask for opinions on job
Search for drug rehab center
Ask for suggestions on rehab centers
Reply

Figure 2: Task division in crowdsourcing location-based services.

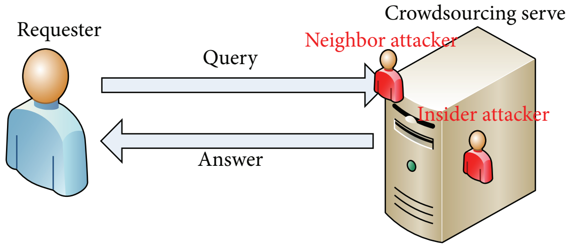


Figure 3: Attacker model.

(4) The "house-job-rehabilitation center" result having the highest crowd rate is sent back to the requestor.

The square blocks in Figure 2 represent machine-based tasks, while the hexagon blocks represent crowd-based ones.

As shown in the above scenario and Figure 2, CrowdSearcher requires user's location as part of the request in order to filter out the crowd not belonging to the area the user is interested in (since geographic location is one of the needed expertise areas). The contributors (crowd) are not aware of the full query but respond only to small portions (e.g., How do you rate this rehab center?) (last hexagon in Figure 2).

As user's location is transmitted, an attacker who is located just before (or within) the crowdsourcing server can capture the full query containing the requester's current location as shown in Figure 3. The attacker model is discussed in Section 7.

It is easy to relate user's identity to the transmitted location and in turn to the request. This identity-query disclosure reveals private information about the user's interests, affiliations, and future plans and possibly helps in tracking him. In our case, the attacker can easily identify the requester and deduce that he or somebody in his small family is a drug addict. The attacker can sell this information to local drug dealers who can target the addict after leaving the rehabilitation center. The relapse rate (return back to addiction after rehabilitation) is very high (40%–60%) [12, 13] which makes the postaddict an easy prey.

Crowdsourced LBS natively compromises the privacy of all its users, because it uses the same security mechanisms utilized by legacy LBS which contains well-known privacy-breaching vulnerabilities. Research efforts were exerted to

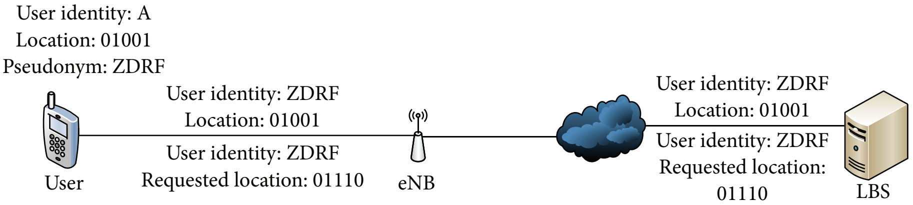


FIGURE 4: Anonymization using pseudonym.

ensure the privacy of legacy LBS users, but all failed to prevent identity-query disclosure. To the best of our knowledge, none has taken advantage of crowdsourced LBS's mobile Cloud nature to offer enhanced privacy.

Matching user's location to his queries has severe security consequences, since location can be related to identity [14] and queries can be related to user preferences, interests, ideas, and beliefs. Being able to match user's identity to preference can be used for customer profiling and behavior expectancy. Tyrant governments would be interested in matching queries and posts countering their regimes to the identity and location of the activists. Safeguarding this match is crucial to maintain user privacy and sometimes user safety.

We are going to show, in this paper, that all the advanced privacy notions, found in the literature, used for anonymization are not suitable for LBS. We are going to show also that the suitable privacy notions have proven vulnerabilities. We can then deduce that location privacy in crowdsourced LBS is not met and could be easily breached; thus, the need for a new privacy model is inevitable. Finally, we are going to propose a new privacy model for crowdsourced LBS based on its mobile Cloud nature.

The remainder of this paper is organized as follows. Section 2 presents background information to make this paper self-contained. Section 3 surveys the latest and most advanced privacy notions found in the literature and proves that none is suitable for crowdsourced LBS except K-anonymity. Sections 4, 5, and 6 discuss the "frequency attack" we developed against K-anonymity. Section 4 shows the mathematical model behind K-anonymity constraint which is used to evaluate the privacy level of different location privacy preserving mechanisms (LPPMs). Section 5 proposes "frequency attack" which is used to exploit the most secure LPPM "footprints." Section 6 shows the simulation results of "frequency attack" to evaluate the privacy breach level.

A new solution for this privacy issue is proposed in Section 7 by changing the computational environment. Finally, Section 8 concludes the paper.

2. Related Work

In this section, we present legacy LBS, which is the prerequisite to understand crowdsourced LBS architecture and the used privacy preserving mechanisms. We also present the used anonymization and privacy notion (K-anonymity). We then survey the privacy requirements that should be maintained by location privacy preserving mechanisms which are also surveyed. At the end of this section, we survey the attacks on K-anonymity to show its vulnerabilities.

2.1. Legacy Location-Based Services and K-Anonymity.

Location-based service is a computer-level online service that utilizes the user's current position as a critical input for the application providing this service [15]. Location coordinates can be delivered through GPS equipped mobile devices [16] or through the mobile user's operator. If a multilateration positioning technique is used in the serving network, then the exact location of the user can be specified. If multilateration is not used, then only the distance to the serving eNB (evolved Node B) is delivered in addition to the eNB's coordinates; in other words, the user knows that it belongs to the circumference of a circle centered at serving eNB; its radius is the user's distance to the center. Note that

(i) location-dependent query is a user triggered request to a location-based service;

(ii) nearest neighbor query is a location-dependent query requesting the address of the nearest point of interest.

Before being able to request any information from location-based services, the mobile user has to update his location by sending the coordinates to the LBS server, which in turn replies with the requested information. Security of the location-related information transmitted over the air channel is considered to be outside the scope of this paper due to the implemented confidentiality and integrity protection at the Access Stratum (AS) layer. We will only consider last mile eavesdropping (between PDN Gateway (P-GW) and LBS server), carried out by outside attackers (neighbor attacks) or the service providers themselves (insider attacks). Capturing insecure identities and location information allows the attacker to breach the user's privacy by being able to know if the user is in a certain area and where precisely he is.

Anonymization was proposed using pseudonym identities, as a cost-effective way to ensure identity privacy. Identity anonymization is implemented at the mobile level, where a pseudonym is generated to replace the username in the LBS query as shown in Figure 4. Pseudonym anonymization failed to ensure the required degree of anonymity [17–19] because each user has a limited number of restricted areas which are known by the attacker and allow him to create a predetermined victim behavior. These restricted areas are places visited regularly by the victim, such as home, office, and home-office road. In other words, the anonymizing technique is vulnerable to correlation attacks. Although the username is anonymized, other remaining attributes called quasi-identifiers can still be mapped to individuals (e.g., age, sex, and city [16, 20, 21]).

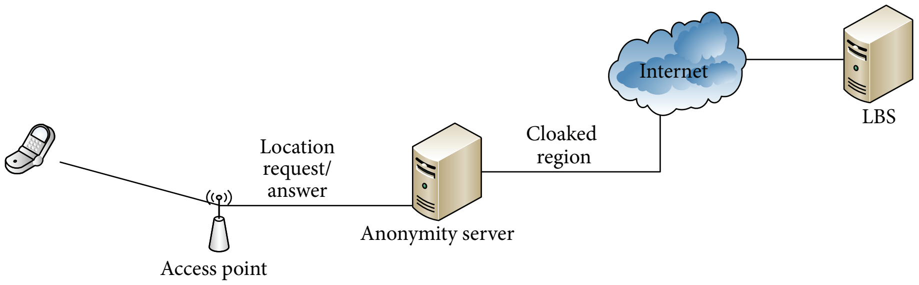


Figure 5: Anonymization using K-anonymity.

The scenario behind Figure 4 is that user A currently found at location 01001 is interested in finding a certain service which is returned by the LBS server as located at 01110.

K-anonymity is another proposed method to ensure location privacy, where a location-dependent query is considered private if the attacker is able to identify the requester with probability less than $1/K$ [22]. K is a threshold required by the user [18]. This method uses an intermediate trusted server called anonymizer as shown in Figure 5.

Every subscribed user has to register with a trusted anonymity server (anonymizer) by updating its identity and location. This anonymizer maintains an updated database of user's current location. Each triggered query passes by the anonymizer; it replaces the user's location by a cloaking region (CR) and forwards the request to LBS server. CR contains, in addition to the real user, $K - 1$ users belonging to its neighborhood. The attacker knows that one within this vicinity has requested this query, but the probability of identifying the right user equals $1/K$. The LBS server replies with a list containing the identity of each user from CR with its corresponding point of interest (POI) (i.e., query's answer). The anonymizer then filters the POI corresponding to the real user and passes it to the requester [17, 18]. K-anonymity also has drawbacks [17, 18]:

(i) The anonymizer is considered a single point of failure and bottleneck; thus, the probability of full outage is higher than that in pseudonym anonymization.

(ii) Malicious users can be physically located near the victim; thus, the anonymizer will add these users in its CR. Since the eavesdropper knows the malicious users, it can predict the user's identity with probability $> 1/K$.

(iii) K-anonymity is also vulnerable to correlation attacks.

(iv) K-anonymity's security level is directly proportional to the frequency of users' location update. It is not practical and scalable to request location updates periodically from all users.

(v) It is more profitable for idle users not to update their location, since moving from LTE_IDLE to LTE_ACTIVE to send this update message will cause higher battery consumption and excess signaling on core level.

(vi) Generated core traffic is $K - 1$ times more than what is needed.

(vii) Only identity and identity-query privacy are ensured but not location privacy.

(viii) It is difficult to support continuous LBS.

Crowdsourcing applications differ from legacy LBSs in the resolution mechanism since one or more employees recruited from the crowd will participate in answering the query, but user's interface and query mechanism (where the privacy mechanisms lay) are slightly changed. This makes the privacy requirements and location privacy preserving mechanism that are used in legacy LBS remain valid in crowdsourcing applications.

The privacy requirements, applicable for legacy and crowdsourced LBS, are surveyed in the next subsection.

2.2. Privacy Requirements of Crowdsourced LBS Applications.

Every crowdsourced LBS should maintain a set of privacy requirements in order to be considered privacy preserving. These requirements are used to prevent the disclosure of sensitive information that could be used as part of more complex attacks. The attacks range from tracking the user to profiling his attitudes and interests. LBS's privacy requirements are as follows:

(i) Location privacy: in crowdsourced LBS, location-identity relationship is relatively easy to be exposed using correlation attacks [14].

(ii) Identity privacy: exposing a user's nonrepudiated identity can expose him to location tracking by exploiting the well-known Evolved Packet System (EPS)/Universal Mobile Telecommunication System (UMTS) tracking vulnerabilities [23, 24].

(iii) Identity-query privacy: LBS requests have no system-wide effect (i.e., wrong information will affect the user's request only without having side effects on other users). Identity traceback is not required.

Location privacy preserving mechanisms (LPPMs) used by LBS applications are surveyed in the next subsection.

2.3. Location Privacy Preserving Mechanisms.

In this subsection, we survey the LPPMs that were proposed for legacy LBS [14] which are as follows:

(i) Anonymization and obfuscation: anonymization is replacing the username part of the LBS query with a pseudonym; obfuscation is responsible for distorting the LBS query's second part (location). Various pseudonym generation mechanisms are found, such as Mix zones [14].

(ii) Private Information Retrieval (PIR): both request and reply are encrypted, leaving the server unable to identify the sender or the request. This mechanism is suitable for legacy LBS but impractical in crowdsourcing, since the search engines need to send the crowd a plaintext job.

(iii) Feeling-based approach: this is achieved by identifying a public region and sending a request with disclosed location that must be at least as popular as that space.

2.4. Attacks on K-Anonymity. In this subsection, we survey the attacks on K-anonymity. To do so, the following terms should be defined first:

(i) Quasi-identifiers: a group of attributes that can uniquely identify an entity (e.g., age, ZIP code).

(ii) Sensitive attributes: information which results in privacy breaching if matched to an entity.

The term "cloaking region (CR)" used in Section 2 is identical to the term "class" used by the authors of [25, 26] who defined it as "a set of records that are indistinguishable from each other with respect to certain identifying attributes" [25]. We will only use CR in this paper to maintain coherence.

Several attacks on K-anonymity have been discussed in the literature, which are as follows:

(i) Selfish behavior based attack: the authors of [27] showed the behavior of selfish users in "anonymization and obfuscation" LPPM and its effect on privacy protection. In the areas where the number of LBS users is less than "k," a requester can generate dummy users so that its CR can again contain "k" users (real and dummy) in order to achieve K-anonymity. Other selfish users (free riders) can benefit from the generated dummy users to obfuscate their requests. The cost of generating dummy users is not distributed fairly among the users benefiting from this LPPM.

(ii) Homogeneity attack [25, 26]: this attack takes advantage of the fact that "K-anonymity creates groups that leak information due to lack of diversity in the sensitive attribute" [25]. The attack expects that the searched attribute (sensitive attribute) is the same for all the users within the same CR. An example showing this attack is implemented as follows: consider the medical records of a certain population shown in Table 1 and its anonymized version published in Table 2 [25]. Alice tries to find the medical situation of Bob who is included in Table 1. Bob is a 31-year-old American male who lives in ZIP code 13053. Alice can deduce that Bob has cancer since all the members belonging to his group (CR = {9, 10, 11, 12}) have the same disease.

Table 1: Medical records of a sample population.

	Nonsensitive (quasi-identifiers)			Sensitive
	ZIP code	Age	Nationality	Situation
1	13053	28	Russian	Heart disease
2	13068	29	American	Heart disease
3	13068	21	Japanese	Viral infection
4	13053	23	American	Viral infection
5	14853	50	Indian	Cancer
6	14853	55	Russian	Heart disease
7	14850	47	American	Viral infection
8	14850	49	American	Viral infection
9	13053	31	American	Cancer
10	13053	37	Indian	Cancer
11	13068	36	Japanese	Cancer
12	13068	35	American	Cancer

(iii) Background knowledge attack: this attack takes advantage of the fact that "K-anonymity does not protect against attacks based on background knowledge" [25] which are not included in the offered information. An example showing this attack is implemented as follows: Alice tries to know the medical situation of her friend Umeko who is admitted to the same hospital as Bob, so his records are also shown in Table 1. Umeko is a 21-year-old Japanese female who currently lives in ZIP code 13068 thus belonging to the first CR $= \{1, 2, 3, 4\}$. Alice can deduce that Umeko has either viral infection or heart disease. It is well known that the Japanese have an extremely low incidence of heart disease; thus, Alice is nearly sure that Umeko has a viral infection.

As shown previously, K-anonymity has various weaknesses and it would be interesting if we are able to replace it. Advanced privacy notions have been proposed to replace K-anonymity for microdata publishing. It looks tempting to adopt these notions into LBS. In the coming section, we are going to show that none of the advanced privacy notions found in the literature except for K-anonymity is suitable for crowdsourced location-based services.

3. K-Anonymity, l-Diversity, and t-Closeness

K-anonymity is considered not enough to ensure the privacy needed for microdata publishing. Microdata are tables that contain unaggregated information which include medical, voter registration, census, and customer data. They are usually used as a valuable source of information for the allocation of public funds, medical research, and trend analysis. Table 1 is a sample microdata table, while Table 2 is an anonymized microdata table. Other stronger privacy notions were proposed in the literature which are l-diversity [25] and t-closeness [26, 28].

The "l-diversity" privacy notion states that a CR is considered to have l-diversity if there are at least l "well-represented" values [25]. "Well-represented" can be interpreted as follows:

Table 2: Anonymized records of the sample population.

	Nonsensitive (quasi-identifiers)			Sensitive
	ZIP code	Age	Nationality	Situation
1	130**	<30	*	Heart disease
2	130**	<30	*	Heart disease
3	130**	<30	*	Viral infection
4	130**	<30	*	Viral infection
5	1485*	>40	*	Cancer
6	1485*	>40	*	Heart disease
7	1485*	>40	*	Viral infection
8	1485*	>40	*	Viral infection
9	130**	3*	*	Cancer
10	130**	3*	*	Cancer
11	130**	3*	*	Cancer
12	130**	3*	*	Cancer

* in the ZIP code column refers to missing data; ** means that 2 digits are anonymized; * in the Nationality column is used to identify an anonymized nationality.

Table 3: Data stored at the LBS anonymizer.

Quasi-identifiers			Sensitive data
Cloaking region	Location	Query	True sender?
CR1	Location 1	Query 1	No
	Location 2		No
	Location 3		Yes
CR2	Location 4	Query 2	No
	Location 5		Yes
	Location 6		No

Table 4: Data captured by the adversary.

Quasi-identifiers			Sensitive data
Cloaking region	Location	Query	True sender?
CR1	Location 1	Query 1	*
	Location 2		*
	Location 3		*
CR2	Location 4	Query 2	*
	Location 5		*
	Location 6		*

* refers to an anonymized Boolean value (true or false).

(i) Distinct l-diversity: it contains at least l distinct values.

(ii) Entropy l-diversity: entropy of the values that occurred is greater than or equal to $\log l$.

(iii) Recursive l-diversity: the most frequent value does not appear too much, and the less frequent value does not appear too rarely.

A table is considered to have l-diversity if all its CRs have l-diversity. Various attacks have been proposed against this privacy notion which results in privacy disclosure [26]. These attacks are as follows:

(i) Skewness attack [26]: skewed distributions could satisfy l-diversity but still do not prevent attribute disclosure. This happens when the frequency of occurrence of a certain sensitive attribute inside a CR differs from its frequency in the whole table (population). Consider the drug addiction rate in a certain population to be 1%. One of the classes has an addiction rate of 50%. This class is considered attribute disclosing since a member of this class is identified as addict with a high probability.

(ii) Similarity attack [26]: distinct but semantically similar sensitive values could satisfy l-diversity but mean the same for the attacker. If the sensitive information shown in a table contains cocaine addict, heroin addict, marijuana addict, and so forth as distinct entries, it could be considered satisfying l-diversity, but for a drug dealer all these entries are considered as a target.

The "t-closeness" concept is another privacy notion that differentiates between the information gained about the whole population and that about specific individuals [26]. As the first gain is tolerated and motivated, the second is considered privacy breaching. A class is considered to have "t-closeness" if the distance between the distribution of a sensitive attribute in the studied class and the distribution in the whole table (population) is less than or equal to a threshold t. The distance is measured using Earth Mover's Distance (EMD).

It is shown in [26] that "t-closeness" ensures higher privacy than "l-diversity."

3.1. *K-Anonymity versus l-Diversity and t-Closeness.*

The difference in nature between microdata tables and LBS CR captured data is the main factor that prevents the implementation of "l-diversity" and "t-closeness" as privacy metrics in LPPMs.

Anonymizing microdata keeps the sensitive attributes intact and hides parts of the quasi-identifiers to prevent uniquely identifying an entity. An entity can only be identified as an unknown record in a class. "l-diversity" and "t-closeness" try to keep the relationship between the entities of a class and its sensitive attributes as private as possible using the techniques discussed previously.

LBS anonymization tries, on the contrary, to hide the sensitive data instead of generalizing the quasi-identifiers which are needed by the location-based server (location, expertise, and query) to be able to respond to the user's query. Table 3 represents the LBS data stored in the anonymizer.

The anonymized data that are transmitted to the LBS server and could be captured by the adversary are shown in Table 4.

As can be seen in the microdata anonymization in Tables 1 and 2 and LBS anonymization in Tables 3 and 4, the latter focuses on hiding the sensitive data which totally negates the aim behind publishing microdata.

Based on the above discussion, we can deduce that l-diversity and t-closeness are not suitable for LBS anonymization; thus, K-anonymity remains the best suitable privacy notion suitable for such services.

We have shown in the previous sections that LBS applications are in need of privacy preserving mechanisms based on valid privacy notions. We have also shown that the already implemented privacy notion, K-anonymity, could be exploited using various attacks, but stricter privacy notions are not suitable. We can conclude that LPPMs using K-anonymity are the best available solutions. To prove the need for a novel privacy concept, we will present in the coming sections a new attack on K-anonymity that we have developed [29].

4. Mathematical Model

We will prove, in this section, that K-anonymity is private if "zero prior knowledge" is assumed and the adversary could not perform the attack for a long continuous period.

Let $\{x_1, x_2, \ldots, x_n\}$ be the set of users found in a CR, where "n" is its size and $n \geq 1$. Let "x" be our investigated user. If $x \notin \{x_1, x_2, \ldots, x_n\}$, that is, $x \notin \text{CR}$, then x is for sure not the requestor. The probability of a user being added to a CR without being the true requestor is

$$P\left(x \in \text{CR}/\left(x_i = \text{requestor}\ \&\ x \neq x_i\right)\right) = \frac{P\left(x \in \text{CR} \cap x_i = \text{requestor}\right)}{P\left(x_i = \text{requestor}\right)} = \frac{n-1}{N-1}, \tag{1}$$

where N is the number of users in the area controlled by the studied anonymizer where $1 \leq n \leq N$. The probability of a user being added to a CR while being the true requestor is

$$P\left(x \in \text{CR}/\left(x = \text{requestor}\right)\right) = 1. \tag{2}$$

We can deduce that the probability of a user being added to a CR is

$$P\left(x \in \text{CR}\right) = \sum_{i=1}^{N} P\left(x \in \text{CR}/x_i = \text{requestor}\right) \cdot P\left(x_i = \text{requestor}\right). \tag{3}$$

If all the users have the same frequency of sending requests (i.e., x_i are equiprobable, i.e., $P(x_i = \text{requestor}) = 1/N$), we can deduce that

$$P\left(x = \text{requestor}/x \in \text{CR}\right) = \frac{P\left(x = \text{requestor}/x \in \text{CR}\right)}{P\left(x \in \text{CR}\right)} = \frac{(1/N)}{(n/N)} = \frac{1}{n}. \tag{4}$$

Based on the assumption that all users are equiprobable (since the adversary has "zero prior knowledge" and could not perform the attack for a long continuous period, he cannot assume otherwise), K-anonymity is satisfied if the cloaking region size is greater than K; that is, $n \geq K$. We will prove, in the coming section, that the privacy offered by K-anonymity could be breached using our proposed attack (frequency attack) if the adversary is able to perform the attack for a long period even if "zero prior knowledge" is assumed.

5. Frequency Attack

Until now, we have considered that all users are equiprobable, but in reality they are not. Consider an LBS which can be used by a user to retrieve the location of the nearest shop (bakery, Chinese restaurant, etc.). While being home, I am aware of the most important places which I usually access, and I know the bakeries, barbers, supermarkets, and so forth within my town so I need no support in locating these places while on the contrary I can help others. When visiting places I am new to, it will be hard for me to find certain targets (metro, supermarket, etc.) especially when facing language problems. My rate of asking for guidance at these new places is much higher than that while being at my known region. Let comfort zone be a set of areas where a certain user is familiar with and needs no support from the studied LBS in finding places. A user's comfort zone includes his hometown and workplace. Outside this comfort zone, the query frequency of this user increases.

Based on the above discussion, we can easily assume that the users' frequencies are not equal; that is, $\exists i/P(x_i = \text{requestor}) \neq 1/N$.

If a user "i" is to be found in a CR, then either he is the real requestor or he is not and selected from the "$N-1$" remaining users.

If user "i" is the real requestor (i.e., $i = x$) then $P(x \in \text{CR}/x = \text{requestor}) = 1$, since the real requestor should be in the CR; else, he will not receive a valid answer.

If user "i" is not the real requestor (i.e., $i \neq x$), then $P(x \in \text{CR}/x = \text{requestor}) = (n-1)/(N-1)$, that is, one of the "$n-1$" selected users from the "$N-1$" remaining users.

Let $P_i = P(x_i = \text{requestor})$; then

$$\begin{aligned} P\left(x \in \text{CR}\right) &= \sum_{i=1}^{N} P\left(x \in \text{CR}/x_i = \text{requestor}\right) \cdot P\left(x_i = \text{requestor}\right) \\ &= \sum_{\substack{i=1 \\ i \neq x}}^{N} \left(\frac{n-1}{N-1}\right) P_i + (1)\, P_x = \left(\frac{n-1}{N-1}\right) \sum_{\substack{i=1 \\ i \neq x}}^{N} P_i + P_x \end{aligned} \tag{5}$$

Since $\sum_{i=1}^{N} P_i = 1$ then $\sum_{\substack{i=1 \\ i \neq x}}^{N} P_i = 1 - P_x$;

then

$$\begin{aligned} P\left(x \in \text{CR}\right) &= \left(\frac{n-1}{N-1}\right)\left(1 - P_x\right) + P_x \\ &= \frac{n-1}{N-1} + \left(\frac{N-n}{N-1}\right) P_x; \end{aligned} \tag{6}$$

let

$$\begin{aligned} a &= \left(\frac{N-n}{N-1}\right), \\ b &= \left(\frac{n-1}{N-1}\right); \end{aligned} \tag{7}$$

then

$$P\left(x_i \in \text{CR}\right) = aP_i + b. \tag{8}$$

The adversary, even if he has "zero prior knowledge," can collect CRs used by the studied application through a silent passive attack. These captured CRs help the attacker to estimate the requesting frequency of each user using the following procedure.

Let x_{ij} be a random variable representing the occurrence of user "i" in the jth collected CR (CR_j) $\cdot$ $x_{ij} = 1$ if user "i" (x_i) is found in CR_j and $x_{ij} = 0$ otherwise. Considering that the attacker has captured "m" CRs, the average number of occurrences of user "i" in the collected CRs is

$$\overline{X}_i = \frac{1}{m}\sum_{j=1}^{m} x_{ij}. \tag{9}$$

We can estimate the population rate (the real rate of user i) from the sample rate with a confidence interval as follows:

$$\begin{aligned} &P\left(x_i \in \text{CR}\right) \in \left[\overline{X}_i - \alpha S\sqrt{\frac{m-1}{m}},\ \overline{X}_i + \alpha S\sqrt{\frac{m-1}{m}}\right] \Longrightarrow \\ &P\left(x_i \in \text{CR}\right) \\ &\quad \in \begin{cases} \left[\overline{X}_i - t^{\alpha} S\sqrt{\dfrac{m-1}{m}},\ \overline{X}_i + t^{\alpha} S\sqrt{\dfrac{m-1}{m}}\right] & \text{if } m < 30 \\ \left[\overline{X}_i - \dfrac{\alpha S}{\sqrt{m}},\ \overline{X}_i + \dfrac{\alpha S}{\sqrt{m}}\right] & \text{elsewhere.} \end{cases} \end{aligned} \tag{10}$$

"t" follows Student's t-distribution of "$m - 1$" degrees of freedom (using Bayesian prediction) and "α" follows normal distribution (using central limit theorem). We can now deduce P_i since

$$\Longrightarrow P\left(x = \text{requestor}/x \in \text{CR}\right) = \frac{P_x}{\sum_{j\in \text{CR}} P_j} \neq \frac{1}{n}. \tag{11}$$

We have shown, in this section, that even with "zero prior knowledge" the adversary is able to estimate the user's request rate through a passive attack aiming to collect CRs. This attack takes advantage of the fact that not all users have the same request rate; that is, each put a different amount of time utilizing a certain application. User request rate escalates when being outside of his comfort zone which helps in differentiating him from other users with lower request rate. As the duration of this attack increases, the attacker's accuracy increases. Using the above estimations, the adversary is able to identify whether a user found in a CR is the real requestor with a probability different than (greater or less than) "$1/n$" which results in lower entropy (uncertainty) and thus better predictability and this breaches the K-anonymity constraint.

We will show, in the coming section, the entropy and rate of successful prediction of our proposed attack.

6. Simulation Results

In this section, we have simulated, using a specially crafted C# code, 50 users within an anonymizer's controlled zone. Each user has either low or high request rates. Users with high request rate are considered outside their comfort zone and have 10 times more requests than low rate users. The privacy level (entropy and rate of successful requestor prediction) is studied for different CR sizes and for sufficient samples. The traffic factor (high request rate/low request rate = 10) is not proved on any real application but assumed as a particular case.

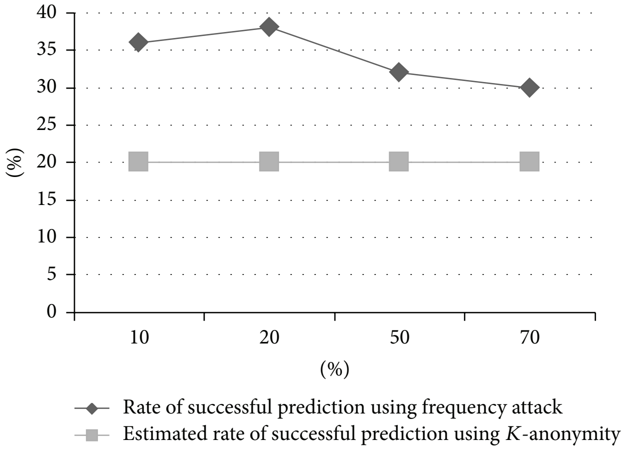


Figure 6: Prediction success rate for CR = 5.

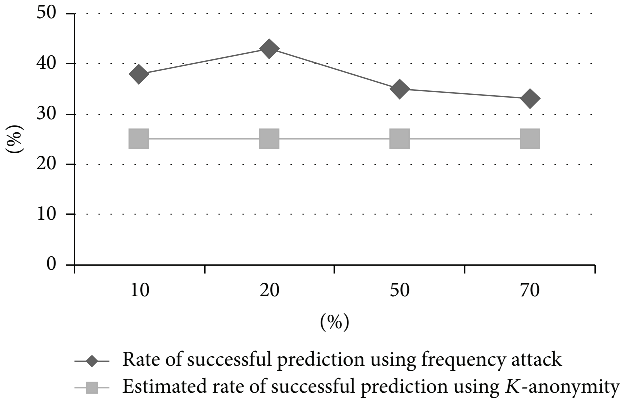


Figure 7: Prediction success rate for CR = 4.

In Figure 6, we can see the estimated rate of successful prediction and the rates achieved by the frequency attack for a cloaking region of size 5. The x-axis represents the proportion of the population with high request rates while the y-axis represents the average probability of predicting the requestor successfully.

It can be shown in Figure 6 that frequency attack has increased the prediction success compared to the estimated rate by (30%-20%)/20% = 50% when 70% of the population are outside their comfort zone to (36%-20%)/20% = 60% when 20% of the population are outside their comfort zone depending on the distribution of the requestors (percentage of users outside their comfort zone). In Figure 7, we can see the estimated rate of successful prediction and the rates achieved by the frequency attack for a cloaking region of size 4. The x-axis and y-axis are similar to those in Figure 6.

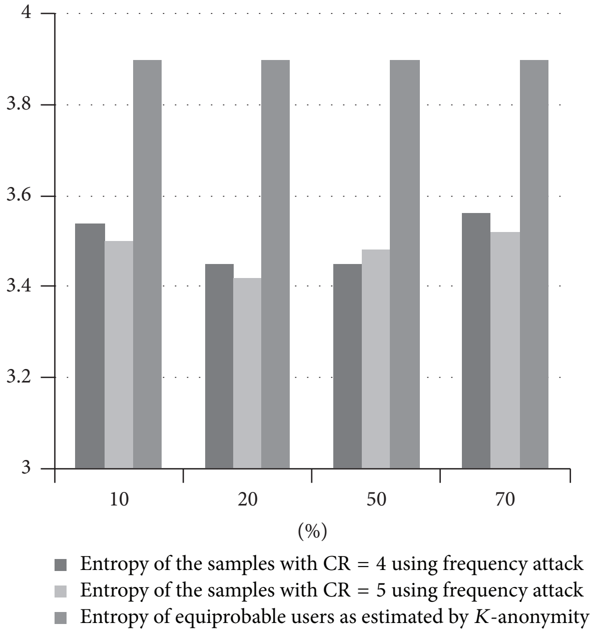


Figure 8: Entropy for frequency attack (CR = 4, CR = 5) and equiprobable users.

It can be shown in Figure 7 that frequency attack has increased the prediction success compared to the estimated rate by (34%-25%)/25% = 30% when 70% of the population are outside their comfort zone to (44%-25%)/25% = 20% when 20% of the population are outside their comfort zone depending on the distribution of the requestors (percentage of users outside their comfort zone). In Figure 8, we can see the entropy (uncertainty) estimated by K-anonymity. The x-axis represents the proportion of the population with high request rates while the y-axis represents mechanism's entropy.

It can be shown in Figure 8 that the entropy (uncertainty of knowing the requestor) is lower when using frequency attack for both CR sizes. This means that the attacker is able to estimate the real requestor with a probability greater than "$1/n$" which leads to a breach in the "K-anonymity" privacy constraint.

We have shown, in this section, that our proposed attack is able to breach K-anonymity with an incremental rate as the duration of this attack increases.

Since we are not able to find any privacy notion that is capable of ensuring elevated user privacy in crowdsourced LBS, we deduce that a solid privacy solution is needed and this will be proposed in the coming section.

7. The Proposed Solution

Ensuring high privacy level in crowdsourced LBS is not trivial in the following environments (attacker models):

(i) Untrusted service provider: in most of the cases, the user has no signed contract with the crowdsourced location-based server; thus, no privacy obligations are held especially when no auditing is taking place.

(ii) Hostile network: crowdsourced LBSs have no means of managing incoming traffic's privacy level other than encryption.

(iii) Untrusted service provider in hostile network: most of the LBSs are untrusted and connected to the Internet (hostile network) but enforce encrypted traffic. This attacker model, which is currently used, results in two attack categories (insider and neighbor) as shown in Figure 3. Figure 9 shows untrusted service provider in a hostile network.

The best privacy can be achieved by a trusted service provider when located within a trusted network. In this case, there is no need to use LPPMs or anonymization since the attacker is neither able to access the sent queries (neighbor attack) nor able to access the log files (insider attack). Any of the LPPMs found in the literature and surveyed in Section 2 generates both processing and traffic overhead; thus, eliminating the need for these mechanisms without affecting the privacy level is definitely an achievement in terms of security and performance.

Since crowdsourced LBS users are in reality mobile users, our proposed solution for this privacy issue is based on our Mobile Cloud Computing architecture named Operator Centric Mobile Cloud Architecture (OCMCA) [30]. We are going next to describe briefly OCMCA.

7.1. Operator Centric Mobile Cloud Architecture. To decrease the delay, cost, and power consumption and increase the privacy, mobility, and scalability compared to other mobile Cloud architectures, we proposed to install a "Cloud server" within the mobile operator's network as shown in Figure 10. Its proposed position leads to the following:

(i) Less delay: the traffic does not need to pass through the Internet to reach the destined server.

(ii) Less cost: the same data contents could be delivered to various users using 1 multicast channel which decreases congestion (scalability) and allowed cheaper pricing models.

(iii) Less power consumption: user's jobs are computed at the Cloud server, which eliminates the need for in-house execution.

(iv) More mobility: a user can maintain a session with the Cloud server as long as he is connected to the mobile network. Our mobile Cloud architecture benefits from the handover mechanisms in place.

(v) More scalability: the Cloud servers are centralized and accessible by all the users.

(vi) More privacy: the Cloud servers are managed by a trusted and secure entity as it will be shown in more detail in the next subsection.

Since users are usually connected with the same operator for long durations, we recommend the user's CSP to federate some resources at the "Cloud server" and then offload the user's applications and environment settings. In this case, all computation-intensive processing is implemented within

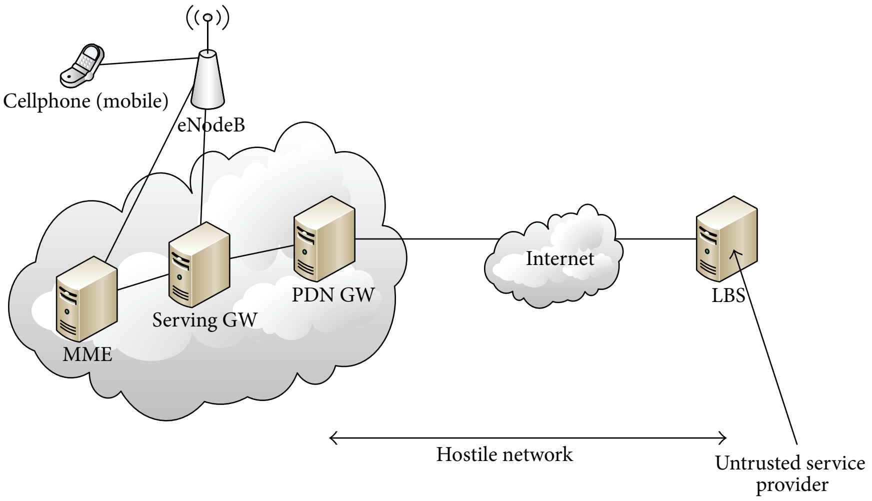


Figure 9: Untrusted service provider in hostile network.

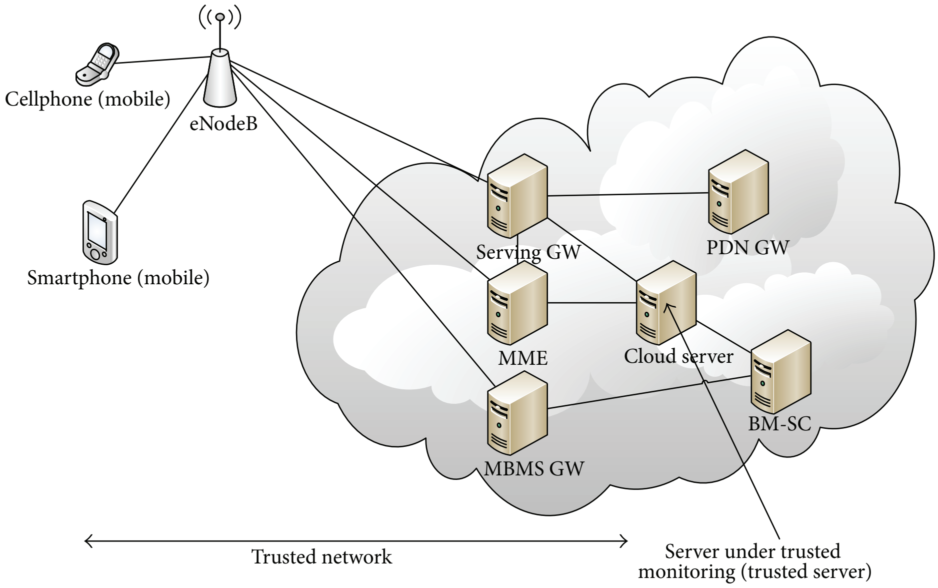


Figure 10: OCMCA.

the mobile operator's network and the terminal generated data are offloaded to the local Cloud without the need to access the Internet (which decreases the cost per bit of the transmitted data). This Cloud should be able to trigger broadcast messages when needed. 3GPP has standardized multicast and broadcast packet transmission in UMTS and broadcast transmission in LTE through a feature named Multimedia Multicast/Broadcast Service (MBMS) which was defined in 3GPP's technical specification as follows:

(i) "MBMS is a point-to-multipoint service in which data is transmitted from a single source entity to multiple recipients" [30].

(ii) Physical broadcasting allows network resources to be shared when transmitting the same data to multiple recipients [30].

(iii) Its architecture ensures efficient usage of radio-network and core-network resources, especially in radio interface [30].

MBMS requires the introduction of two nodes to the Evolved Packet Core (EPC) network [30], which are as follows:

(i) MBMS GW: which is responsible for the connections with content owners, Cloud servers in our case.

(ii) BM-SC (Broadcast Multicast Service Center): which provides a set of functions for MBMS user services.

If MBMS is already enabled in a studied mobile operator, then MBMS GW and BM-SC are already installed. Enabling broadcast in LTE allows the local Cloud, located within the mobile operator's premises, to transmit data efficiently to the users of one or more cells.

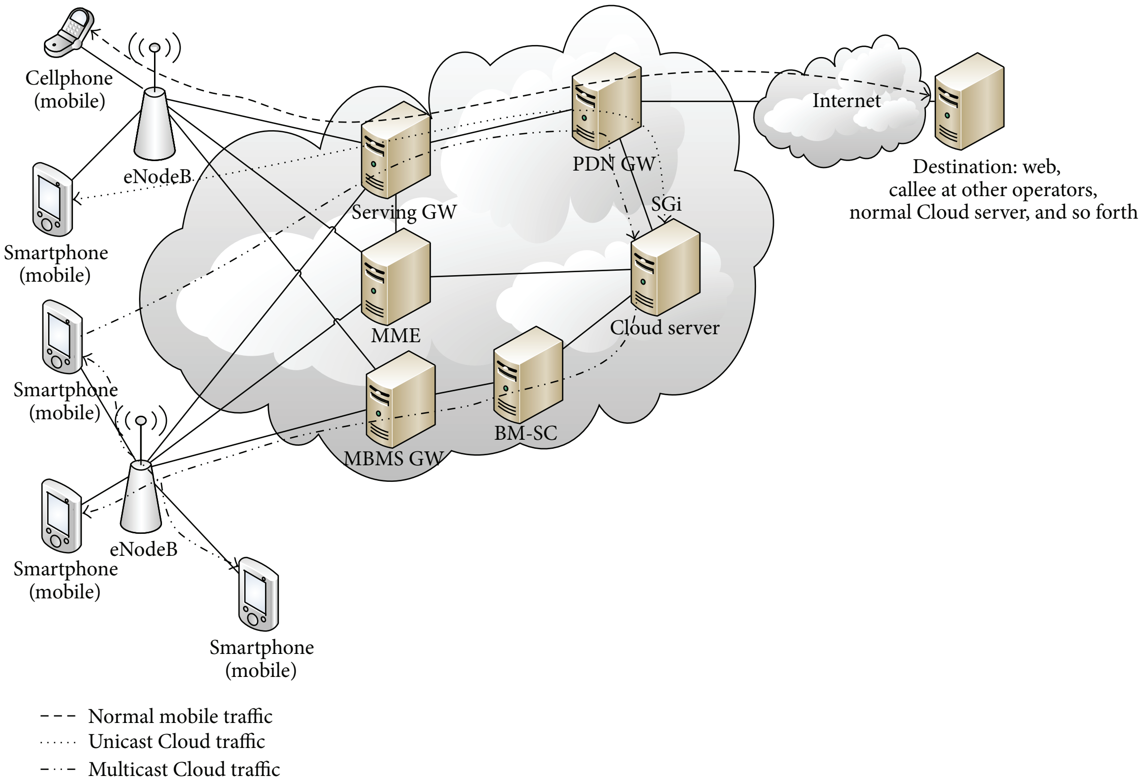


Figure 11: Traffic routes in OCMCA.

The added nodes (MBMS GW, BM-SC, and Cloud server) are transparent to normal mobile traffic. Mobile traffic will pass as follows:

(i) Normal mobile traffic: eNodeB, Serving GW, and Packet Data Network GateWay (PDN GW), shown in Figure 11.

(ii) Unicast Cloud traffic: eNodeB, Serving GW, "Cloud server," Serving GW, and eNodeB, shown in Figure 11.

(iii) Multicast Cloud traffic: eNodeB, Serving GW, "Cloud server," BM-SC, and MBMS GW, shown in Figure 11.

7.2. Privacy Guaranteeing Architecture. One of the "Cloud server's" capabilities is to offer Software-as-a-Service (SaaS) on behalf of application developers. An application developer can request the operator to host his service (application). After reviewing the source code, the operator agrees on hosting this service and signs an SLA with the developer.

A similar protocol is implemented by Apple before adding any application to Apple Store [31]. The application developer trusts Apple for not spreading the source code and for billing (charging every downloaded instance).

In our case, the LBS provider will delegate his application to be provided by the mobile operator on his behalf as shown in Figure 12.

As can be seen in Figure 12, the LBS query does not need to pass in a hostile network which eliminates neighbor attacks. Since the offered service is under the operator's monitoring, the provider becomes trusted and insider attacks become eliminated. The application verification and billing methods suitable for the operator could be used without affecting the proposed solution.

Since LPPMs have been proposed to prevent an attacker who captured LBS queries from identifying the requestor or tracing the users, we can consider that LPPMs are not needed anymore in this scenario, since attackers are not capable of capturing the transmitted requests. Based on this discussion, we consider the following:

(i) User identity privacy is maintained. The operator has access to user's real identity (IMSI, MSISDN, and name) and generates temporary identities (TMSI, GUTI, etc.) to prevent real identity breach. User's identity privacy is maintained since the operator is considered a trusted entity but, in all cases, it already has access to this information; thus, our proposed solution does not offer any new information that the operator could benefit from.

(ii) User location privacy is maintained. The operator already has access to any user's location and does not benefit from additional sensitive information leading to a transparent and strong location privacy solution for the user.

8. Conclusion

User privacy in crowdsourced LBS is unacceptable in its current state. Available location privacy preserving mechanisms result in unnecessary overhead without being able to

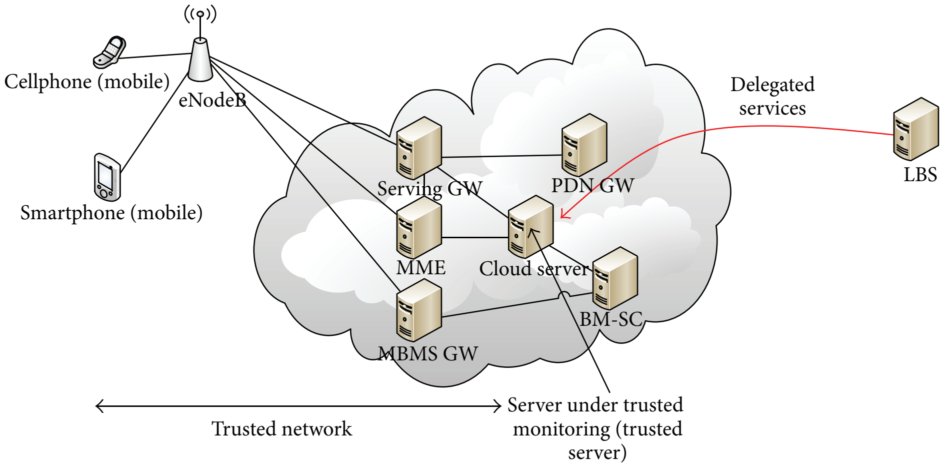


Figure 12: Crowdsourced LBS delegation.

satisfy the needed requirements. We have shown that even the state-of-the-art approach such as K-anonymity is not suitable to fulfill the required privacy needs as it is vulnerable to our defined frequency attack. Thus, a new privacy model is needed and this is what we have proposed in this paper.

The delegation of services into the Cloud server inside the operator's trusted network prevents all the attacks proposed by the attacker model shown in Figure 3.

Competing Interests

The authors declare that they have no competing interests.

References

[1] Google Cloud, https://Cloud.google.com/.

[2] A. Bozzon, M. Brambilla, and S. Ceri, "Answering search queries with CrowdSearcher," in *Proceedings of the 21st Annual Conference on World Wide Web (WWW '12)*, pp. 1009–1018, ACM, Perth, Australia, April 2012.

[3] B. L. Bayus, "Crowdsourcing new product ideas over time: an analysis of the Dell IdeaStorm community," *Management Science*, vol. 59, no. 1, pp. 226–244, 2013.

[4] J. Howe, *Crowdsourcing: How the Power of the Crowd Is Driving the Future of Business*, Random House, New York, NY, USA, 2008.

[5] M. F. Bulut, Y. S. Yilmaz, and M. Demirbas, "Crowdsourcing location-based queries," in *Proceedings of the 9th IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops '11)*, pp. 513–518, Seattle, Wash, USA, March 2011.

[6] D. Horowitz and S. D. Kamvar, "The anatomy of a large-scale social search engine," in *Proceedings of the 19th International World Wide Web Conference (WWW '10)*, pp. 431–440, Raleigh, NC, USA, April 2010.

[7] G. Chatzimilioudis, A. Konstantinidis, C. Laoudias, and D. Zeinalipour-Yazti, "Crowdsourcing with smartphones," *IEEE Internet Computing*, vol. 16, no. 5, pp. 36–44, 2012.

[8] Chacha.com, http://www.chacha.com/.

[9] Foursquare.com, https://foursquare.com/.

[10] W. S. Lasecki, R. Wesley, J. Nichols, A. Kulkarni, J. F. Allen, and J. P. Bigham, "Chorus: a crowd-powered conversational assistant," in *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology (UIST '13)*, pp. 151–162, St Andrews, UK, October 2013.

[11] A. Noulas, S. Scellato, C. Mascolo, and M. Pontil, "An empirical study of geographic user activity patterns in foursquare," in *Proceedings of the International Conference on Weblogs and Social Media*, AAAI, Barcelona, Spain, July 2011.

[12] How effective drug addiction treatment, http://www.drugabuse.gov/publications/principles-drug-addiction-treatment-research-based-guide-third-edition/frequently-asked-questions/how-effective-drug-addiction-treatment.

[13] Drug Rehab Statistics, http://www.futuresofpalmbeach.com/drug-rehab/statistics/.

[14] J. Bou Abdo, H. Chaouchi, and J. Demerjian, "Security in emerging 4G networks," in *Next-Generation Wireless Technologies: 4G and Beyond*, pp. 243–272, Springer, London, UK, 2013.

[15] Location-Based Service, http://en.wikipedia.org/wiki/Location-based_service.

[16] M. E. Nergiz, M. Atzori, and Y. Saygin, "Towards trajectory anonymization: a generalization-based approach," in *Proceedings of the SIGSPATIAL ACM GIS International Workshop on Security and Privacy in GIS and LBS (SPRINGL '08)*, pp. 52–61, November 2008.

[17] S. Steiniger, M. Neun, and A. Edwardes, *Foundations of Location Based Services*, Lecture Notes on LBS, v. 1.0, 2006.

[18] D. Quercia, N. Lathia, F. Calabrese, G. Di Lorenzo, and J. Crowcroft, "Recommending social events from mobile phone location data," in *Proceedings of the IEEE 10th International Conference on Data Mining (ICDM ' 10)*, pp. 971–976, IEEE, Sydney, Australia, 2010.

[19] B. Jiang and X. Yao, "Location-based services and GIS in perspective," *Computers, Environment and Urban Systems*, vol. 30, no. 6, pp. 712–725, 2006.

[20] G. Ghinita, P. Kalnis, A. Khoshgozaran, C. Shahabi, and K.-L. Tan, "Private queries in location based services: anonymizers are not necessary," in *Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD '08)*, pp. 121–132, ACM, 2008.

[21] G. Ghinita, P. Kalnis, M. Kantarcioglu, and E. Bertino, "A hybrid technique for private location-based queries with database

protection," in *Advances in Spatial and Temporal Databases*, N. Mamoulis, T. Seidl, T. B. Pedersen, K. Torp, and I. Assent, Eds., vol. 5644 of *Lecture Notes in Computer Science*, pp. 98–116, Springer, Berlin, Germany, 2009.

[22] L. Sweeney, "*k*-anonymity: a model for protecting privacy," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 10, no. 5, pp. 557–570, 2002.

[23] Generation Partnership Project, 3GPP T TR 33.821 V9.0.0 (2009-06), 3GPP Rationale and Track of Security Decisions in Long Term Evolved (LTE) RAN/3GPP System Architecture Evolution (SAE) (Release 9).

[24] J. Bou Abdo, J. Demerjian, and H. Chaouchi, "Security V/S Qos for LTE authentication and key agreement protocol," *International Journal of Network Security & Its Applications*, vol. 4, no. 5, pp. 71–82, 2012.

[25] A. Machanavajjhala, D. Kifer, J. Gehrke, and M. Venkitasubramaniam, "L-diversity: privacy beyond k-anonymity," *ACM Transactions on Knowledge Discovery from Data*, vol. 1, no. 1, article 3, 2007.

[26] N. Li, T. Li, and S. Venkatasubramanian, "t-closeness: privacy beyond k-anonymity and l-diversity," in *Proceedings of the IEEE 23rd International Conference on Data Engineering*, pp. 106–115, Istanbul, Turkey, April 2007.

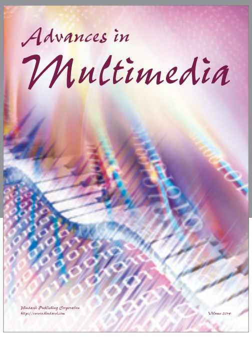
[27] X. Liu, K. Liu, L. Guo, X. Li, and Y. Fang, "A game-theoretic approach for achieving k-anonymity in Location Based Services," in *Proceedings of the 32nd IEEE Conference on Computer Communications (IEEE INFOCOM '13)*, pp. 2985–2993, Turin, Italy, April 2013.

[28] D. Rebollo-Monedero, J. Forné, and J. Domingo-Ferrer, "From t-closeness-like privacy to postrandomization via information theory," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 11, pp. 1623–1636, 2010.

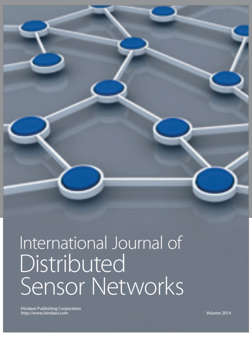
[29] J. B. Abdo, J. Demerjian, H. Chaouchi, T. Atechian, and C. Bassil, "Privacy using mobile cloud computing," in *Proceedings of the 5th International Conference on Digital Information and Communication Technology and Its Applications (DICTAP '15)*, pp. 178–182, IEEE, Beirut, Lebanon, May 2015.

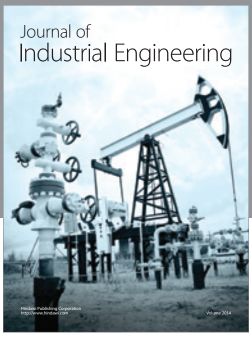
[30] J. B. Abdo, J. Demerjian, H. Chaouchi, K. Barbar, and G. Pujolle, "Operator centric mobile cloud architecture," in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC '14)*, pp. 2982–2987, IEEE, Istanbul, Turkey, April 2014.

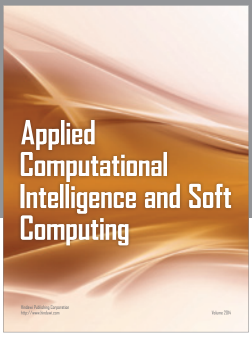
[31] Submitting Your Apple to the Store, https://developer.Apple.com/library/ios/documentation/IDEs/Conceptual/AppDistributionGuide/SubmittingYourApp/SubmittingYourApp.html

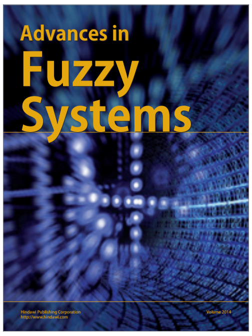

Advances in
Multimedia

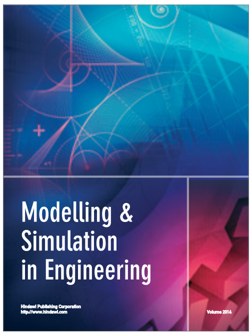
The Scientific
World Journal

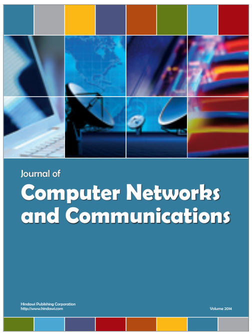

International Journal of
Distributed
Sensor Networks

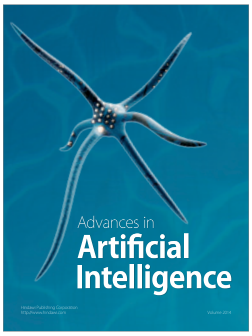

Journal of
Industrial Engineering

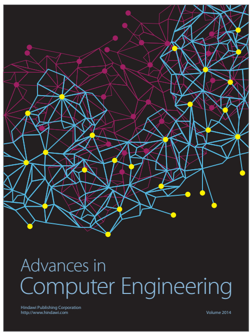

Applied
Computational
Intelligence and Soft
Computing


Advances in
Fuzzy
Systems

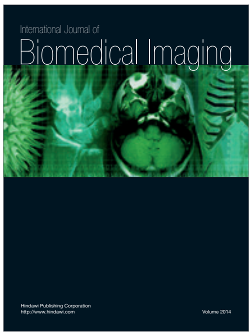

Modelling &
Simulation
in Engineering

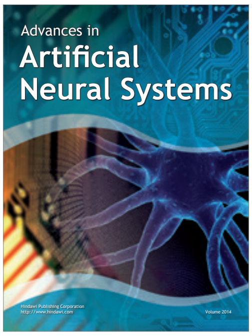

Journal of
Computer Networks
and Communications

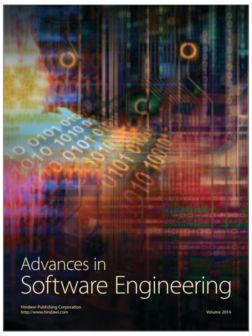

Advances in
Artificial
Intelligence


Advances in
Computer Engineering

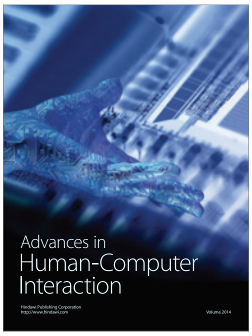
International Journal of
Computer Games
Technology

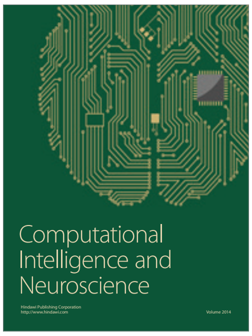

International Journal of
Biomedical Imaging


Advances in
Artificial
Neural Systems


Advances in
Software Engineering

Journal of
Robotics


Advances in
Human-Computer
Interaction


Computational
Intelligence and
Neuroscience

International Journal of
Reconfigurable
Computing

Journal of
Electrical and Computer
Engineering