Journal of Network and Computer Applications 53 (2015) 74–90

Contents lists available at ScienceDirect

Journal of Network and Computer Applications

journal homepage: www.elsevier.com/locate/jnca

Review

A review paper on preserving privacy in mobile environments

S. Arunkumar [a,b,*], M. Srivatsa [c], M. Rajarajan [a]

[a] School of Mathematics Computer Science and Engineering, City University, Northampton Square, London EC1V 0HB, UK
[b] IBM, UK
[c] IBM Research, USA

ARTICLE INFO

Article history:
Received 18 August 2014
Received in revised form
10 December 2014
Accepted 13 January 2015
Available online 14 March 2015

Keywords:
Mobile
Privacy
Security

ABSTRACT

Technology is improving day-by-day and so is the usage of mobile devices. Every activity that would involve manual and paper transactions can now be completed in seconds using your fingertips. On one hand, life has become fairly convenient with the help of mobile devices, whereas on the other hand security of the data and the transactions occurring in the process have been under continuous threat. This paper, re-evaluates the different policies and procedures used for preserving the privacy of sensitive data and device location. Policy languages have been very vital in the mobile environments as they can be extended/used significantly for sending/receiving any data. In the mobile environment users always go to service providers to access various services. Hence, communications between the service providers and mobile handsets needs to be secured. Also, the data access control needs to be in place. A section of this paper will review the communication paths and channels and their related access criteria. This paper is a contribution to the mobile domain, showing the possible attacks related to privacy and the various mechanisms used to preserve the end-user privacy. In addition, it also gives a comparison of the different privacy preserving methods in mobile environments to provide guidance to the readers. Finally, the paper summarizes future research challenges in the area of privacy preservation. This paper examines the 'where' problem and in particular, examines tradeoffs between enforcing location security at a device vs. enforcing location security at an edge location server. This paper also sketches an implementation of location security solution at both the device and the edge location server and presents detailed experiments using real mobility and user profile data sets collected from multiple data sources (taxicabs, Smartphones).

Contents

* Correspondence to: 7 Copse View Close, Basingstoker RG248EZ, United Kingdom. Tel.: +44 7738313817.
E-mail addresses: saritha.arun@uk.ibm.com (S. Arunkumar), msrivats@us.ibm.com (M. Srivatsa), R.Muttukrishnan@city.ac.uk (M. Rajarajan).

http://dx.doi.org/10.1016/j.jnca.2015.01.008
1084-8045/

1. Introduction

Mobile devices have become an important tool in modern day communication. Mobile and other handheld devices such as ipads and tablet PCs have overtaken laptops and desktops and hence there has been an increasing research interest in the area of mobile computing in recent years. This includes areas such as quality of communication, usability and the overall end-to-end data security in day-to-day mobile transactions. Today's mobile devices continuously connect to different service providers for day-to-day online activities such as online purchases, online banking, social networking and endless web surfing. In addition to this, devices could be connecting to the service providers to receive or send sensitive information. At the Service Provider end, the data would be stored and Service Provider would only hand-over the data if it confirms that the person requesting it is authorized to receive the information. The exchange of data from one end of the network to the other is a major challenge due to the mishandling of the data by a malicious user. Hence the confidentiality and integrity of the data needs to be protected either by transforming the sensitive information into a non-readable format or by converting it into a cipher text.

Mobile environments are always prone to various security vulnerabilities. A number of papers have been written to highlight the various threats and problems due to the large volume of transactions occurring in the mobile environments (Jamaluddin et al., 2004; Liu et al., 2009). A very popular attack on the mobile environment is the man-in-the-middle attack. Every bit of data that comes into the mobile device and goes out of the mobile device can be assumed to be sniffed by a malicious user. The information can be assumed to be sniffed by the man-in-the-middle and manipulated in order to retrieve the sensitive information. Protecting the information that is being exchanged between the mobile devices is a major challenge and this paper will discuss some of the techniques that can be employed to mitigate the man-in-the-middle attack. The attack discussed above includes a number of attacks such as man-in-the-middle, sniffing and privacy related attacks. Another attack that is described by some of the researches is based on the cross service attack on the mobile devices (Mulliner et al., 2006). Cross service attacks can occur while you are browsing from your mobile handset sitting in a shop with wireless connectivity. The malicious user would be monitoring the new connections to the wireless network and using an exploit published previously he gains access to the phone. Mulliner et al. (2006) describes in detail the proof-of-concept to show the attack and also discusses about the way in which the vulnerability can be exploited.

With the increasing availability of mobile devices, there is a growing demand for location-based applications. In response to such a user demand, various location-based services have been emerging recently (Beresford and Stajano, 2004; ⟨http://www.mobiloco.de⟩).

A very interesting type of attack that has been popular in mobile and smartphones is the video based attack. All 3G smartphones have the bluetooth, camera and video capabilities and hence is prone to video based vulnerabilities. Xu et al.(2009) have come up with stealthy video capturing software that captures the user behavior patterns/data without the owner's knowledge. It then sends the collected information into a remote device. This attack is executed in such a way that the device owner is unaware of the devices activities. Stealthy Video Capturer (SVC) is a spyware that works very well in all 3G smartphones. All it needs is the 3G connectivity and the video recording capability. This works based on the Windows mobile 5.0/6.0 platforms and it uses the relevant API's for it's functioning. The three main components of this spyware are: Video capture, triggering algorithm and file sending. The video capture as the name suggests captures the video without the knowledge of the mobile user. The triggering algorithm identifies the precise time to turn on the video capturing process and passes on the video information. Finally the file sending flow is responsible for sending the recorded video to a remote device. The video is compressed using mobile phone's video compression techniques before it is being sent to the remote location. They also discuss the injection method used in SVC. As most users today download a lot of games from the Internet, the authors in Xu et al. (2009) found a way of injecting the Trojan using a game and to achieve this they used the tic-tac-toe game. In this case the owner of the mobile device downloads the game and is content that he has just received a new game. However, he is totally unaware of the SVC that has also been downloaded together with the game. It can also be noted that the CPU, memory and other details of the phone that needs to be looked at before the triggering algorithm captures a video. The authors also comment that the malware is resistant to all the existing antivirus tools as it is a new type of vulnerability. The key factor contributing to the success of SVC is due to the fact that there is no efficient management policy for system APIs security for Windows Mobile.

In the mobile environment, it is quite common to have a man-in-the-middle trying to sniff at the information being passed between the mobile device and the service providers (Mulliner et al., 2006). Therefore it is crucial to have data access control mechanisms in place.

It would be interesting to highlight the importance of European data protection guidelines that has recently undergone revisions to include the privacy of individual's data and personally identifiable information (PIIs). Some of the notable changes include explicit consent from the user when data is being shared with other third party service providers. More transparency about the way in which the data is handled is another important change to the European Data Privacy Directive. The reform also includes the mandate for complete accountability and responsibility of the service provider when personal data is being processed (⟨http://ec.europa.eu/justice/newsroom/data-protection/news/120125_en.htm⟩).

This review paper mainly covers the various methods used for preserving user privacy. Hence, it presents a detailed review of many methodologies before moving onto the open research problems in the various solutions described. It then moves onto

discuss “where to enforce security” and shows a novel approach to enforce the location security.

The rest of the paper is organized as follows: Section 2 covers the related work. Section 3 details the privacy related attacks. Section 4 describes the different mechanisms and the classification of preserving privacy in mobile environments. Sections 5–13 discuss the different methods of preserving privacy in various mobile environments. Section 14 compares the different mechanisms and the drawbacks in the existing solutions. It also discusses the open security problems and future research directions. Section 15 describes the location privacy and “where to enforce location privacy”. Section 16 introduces Mobile Microcloud. Section 17 describes the security matrix. Section 18 explains the android implementation for location privacy. Section 19 includes the conclusion and followed by acknowledgment.

2. Related work

A number of papers related to mobile environments and its vulnerabilities have been published in the recent past. Sniffing attacks have been talked about in Cai et al. (2009). They explore the vulnerability where attackers snoop on users by sniffing on their mobile phone sensors, such as the microphone, camera, and GPS receiver.

Schlegel et al. (2011) discuss about Soundcomber, which is a stealthy Trojan with innocuous permissions that can sense the context of its audible surroundings to target and extract a very small amount of high-value data. As sensor-rich smartphones become more ubiquitous, sensory malware has the potential to breach the privacy of individuals at mass scales. There have been a number of different papers concentrating on the different vulnerabilities of mobile devices and how the operating system in the device allows users to control access to sensitive information including location, camera images, and contacts. In Enck et al. (2010) authors have introduced TaintDroid, which operates as an efficient system wide information flow-tracking tool. This tool has the capability of tracking multiple sources of sensitive data. The authors also studied the behavior of thirty popular third party applications chosen at random from Android marketplace and concluded that two-thirds of those applications display suspicious handling of sensitive data.

A paper is dedicated to the mobile phone vulnerabilities, which talks about the different malwares that are targeted on the mobile devices (Jamaluddin et al., 2004). The paper details on how some of the malwares can be implemented easily in order to make the mobile phones vulnerable to attacks. Preventing the cell phones from malicious users or infiltrators is very important and there have been a number of research papers concentrating on the same. In Liu et al. (2009), VirusMeter is detailed which detects existence of malware with abnormal power consumption. VirusMeter relies on a concise lightweight user-centric power model and aims to detect mobile malware in two modes: While the real-time detection mode provides immediate detection, running VirusMeter under the battery-charging mode can further improve the detection accuracy without concerns about resource consumption. Using real-world malware the authors have experimentally shown that VirusMeter can effectively and efficiently detect their existence.

In Lia et al. (2006) authors adapted a special and feasible method, blind signature, to generate an authorized anonymous ID that replaces the real ID of an authorized mobile device. They presented a two-phase protocol to address location privacy, however, did not consider that the randomness introduced during the blinding phase can be removed easily. They also prove that the administrator can link real ID with authorized anonymous ID. In addition to this they propose an improved registration and reconfusion protocol using the same cryptographic technique, blind signature based on bilinear pairings.

A considerable amount of research work has been carried out in the area of location-based applications. In Chen and Wang (2011), authors propose a security model for location based services using outsourced databases and demonstrate how one can use distributed storage and international mobile subscriber identity (IMSI) as user identification to secure the location data. In He et al. (2004), the authors investigated the problem of protecting location privacy of mobile users in the setting of ubiquitous computing. They find it challenging, as there are various requests that are forced by the organization and the users. In papers He et al. (2004), Qi et al. (2004) authors proposed an authorized-anonymous-ID based scheme, which is used to replace the real ID of an authorized mobile device. With authorized-anonymous-IDs, they also designed an architecture that is able to provide the mobile users with complete control over their location privacy and still allowing the organization to authenticate the mobile users. In Zhong et al. (2005), the authors have designed novel protocols to provide location-based services, which do not require a user to trust a third party. They also analyzed a class of location-based services that do not directly transfer user locations.

Barkhuus and Dey (2003) discuss users' concerns about the location-based services that would disclose their location and in turn user's privacy. In this paper, the authors have presented two types of location-based services, location-tracking and position-aware services. They have shown a case study that examines user's concern for privacy in relation to location-based services and compared people's perceived usefulness of the two types of services. The paper concludes that the concerns are more when third parties are tracking a user's location.

Location based services with privacy as the main concern has been described in Beresford and Stajano (2004), Mobiloco et al. (⟨http://www.mobiloco.de⟩), Escudero-Pascual et al. (2002), Escudero et al. (2001), Beresford (2005), Ghinita1 et al. (2007) and Gedik et al. (2005). In Beresford and Stajano (2004), authors have refined the mix zone model, describing a quantifiable metric of location privacy from the point of view of the attacker. In Minch (2004), the author discusses the issues in the location-aware mobile devices in context by addressing the basic technology issues involved. He also discusses issues that are possible and not possible in the future. Further he outlines privacy issues that arise from the conjunction of technical feasibility and government/marketplace activities that might use location information. In this paper a representative sample of important issues is enumerated and discussed. Regulation is then discussed as a broader term covering the various entities and agencies that might structure and regulate the use of location information and provide the appropriate levels of privacy protection to constituents while promoting appropriate advances in new products and services.

Other challenges such as user privacy are also important in ubiquitous environments. Privacy related efforts have been made in the past (Brar and Kay, 2004). Research has been carried out around privacy awareness systems that allow certain privileges to data collectors (Langheinrich, 2002). Karyda and Gritzalis (2009) listed some of the challenges in this area and the future research directions.

3. Possible privacy related attacks

There has been a number of privacy related attacks that have come into existence today. One of the attacks is a sensor sniffing attack in which it assumes that the threat model is where the attackers are able to install malicious software onto the devices. This can be done by exploiting the software vulnerabilities or by tricking to install untrusted code. It is also assumed that the

attacker has no physical access to the device but can receive the sensor date through voice or data channels. More details about this can be found in the paper by Cai et al. (2009).

A number of viruses have been created to exploit the vulnerabilities that exist on today's mobile devices. One of the viruses, which originated in Spain, sends text messages to random mobile phone numbers (Jamaluddin et al., 2004). As mobile phones become more and more intelligent the attacks against them will keep increasing. A number of vulnerabilities have been exploited using the Bluetooth capability of mobile devices leading to exposure of personal data (Jamaluddin et al., 2004). Another potential attack that has been in existence is stealing user's personal data and downloading it without the consent of the mobile owner (Jamaluddin et al., 2004).

Another area of attacks relating to the privacy of mobile devices is through Trojan applications. A number of Trojan applications have been created which gets installed onto the mobile device and starts exploiting and misusing the capabilities. Jamaluddin et al. (2004) detail some of the vulnerabilities that are used to exploit mobile phones and the breach of privacy through them. A very recent report highlighted that Google's Android phones are vulnerable to privacy attacks (⟨http://www.bbc.co.uk/news/business-17192234⟩). The vulnerability results from the use of unencrypted wireless networks like Wi-Fi to log into various Google services such as contacts, calendar and services like Picasa. When users request a digital certificate to sign into these services without re-typing the login information, Google's servers relay an authentication token back to the user's phone. This allows the user to be able to be logged into the accounts for 2 weeks without having to re-login. This sounds like a matter of convenience to the user but it has turned out to be a security flaw due to the fact that the authentication token is sent out in plain text. Malicious users can track the unsecured network and capture the authentication token thus allowing access to various services leading to a total breach of privacy.

Recent news has shown that some of Europe's biggest mobile phone companies signed up to new privacy guidelines published by the GSMA (Global System for Mobile communication Association) (⟨http://www.bbc.co.uk/news/technology-17178954⟩).

The body, which represents mobile operators hopes it will help users understand what personal information apps, may access, collect and use. Several companies have said that they are starting to implement the guidelines in apps they produce. The policy's publication follows concern that some apps were using customer data without the permission of the data owner.

Anxieties about smartphone application privacy were raised after the makers of Path and Hipster apps admitted uploading user contact data without explicit consent (⟨http://www.bbc.co.uk/news/technology-17178954⟩) of the data owners.

Twitter also updated its privacy policy over concerns about how its mobile app used address book information. And recent reports have led to similar fears about the way in which some of the applications accessed private information.

IT can be noted that GSMA has provided guidelines to the application developers asking them to respect the privacy of the users (⟨http://www.bbc.co.uk/news/technology-17178954⟩). The guidelines recommend that the users be informed of exactly who would access what information and with whom the information would be shared for what purpose.

One of the recent news also highlighted the fact that companies such as Google fails to meet the European Union's data protection laws (⟨http://www.bbc.co.uk/news/business-17192234⟩). This is of great concern and hence it is very important to use adequate guidelines and policies, which would help in maintaining the privacy of the user and the user's sensitive information.

Android is a core delivery platform providing ubiquitous services for connected smartphone paradigm, thus monetary gains have prompted malware authors to employ various attack vectors to target Android. Due to large increase in unique malware app signature(s) and limited capabilities within Android environment, signature based methods are not sufficient against unseen, cryptographic and transformed code. Researchers have proposed various behavioral approaches to guard the centralized app markets as malware authors are targeting easy-to-reach-user online distribution mechanism. Issues such as malware penetration and stealth techniques exist. Signature based methods can be easily circumvented using code obfuscation necessitating a new signature for each malware variant (Fedler et al., 2013), forcing the anti-malware client to regularly update its signature database.

The research community for malware analysis and detection is currently worked out static and dynamic approaches. Although these approaches can be used independently each one of these techniques comes with its own limitations. There is not a single technical solution that can address all the known vulnerabilities. To tackle wide variety of new malware, a comprehensive evaluation framework incorporating robust static and dynamic methods can be proposed on Android platform. Manual analysis has become infeasible due to the exponential increase in the number of unknown malware samples. Recent Research has proposed an automated, hybrid approach for Android malware analysis.

4. Classification of preserving privacy in mobile environments

The below architecture shows the complete classification of the different techniques used to preserve the privacy in mobile environments. It defines the problems involved as well as the techniques proposed to overcome these shortcomings. The privacy techniques are classified under two main headings: (1) Data privacy and (2) Contextual privacy. Data privacy mainly involves the data that is being transmitted to and from the mobile device. This data could be in the form of a message, text or information. The data could be sensitive information or it could even be a confirmation on some booking that was done for an online shopping.

Figure 1 shows the privacy classifications and within the data privacy section, it shows the two main areas of problems, i.e. the mobile query and the mobile resources. The mobile query could request the service providers for information that could be sensitive in nature. Hence this has always been a problem to understand and hence preserve the privacy of the information.

In addition to this, the data confidentiality is guaranteed through authentication. The other area of classification of privacy is based on the contextual privacy.

Contextual privacy can be further divided into two areas namely location privacy and identity privacy.

When mobile users request for static resources or mobile resources, pseudo–identifiers are sent and location is anonymized. The data that is being transmitted is protected against third party malicious users. Although the information can be assumed at all times to be hijacked by malicious users, malicious users protect the data against unauthorized access. This is achieved by using data access control mechanisms such as P3P policy extension and XACML policies.

These are described in detail in the later sections. Location privacy mainly deals with the location of the requester. In mobile environments, users are frequently requested for their location information when they try to access a new online service. For example, when a user requests for nearby restaurant information from a location-based server, the location based server needs to know the location of the user and hence the location information is normally requested. However, in most of the cases, the user does not want to disclose the location information to arbitrary location

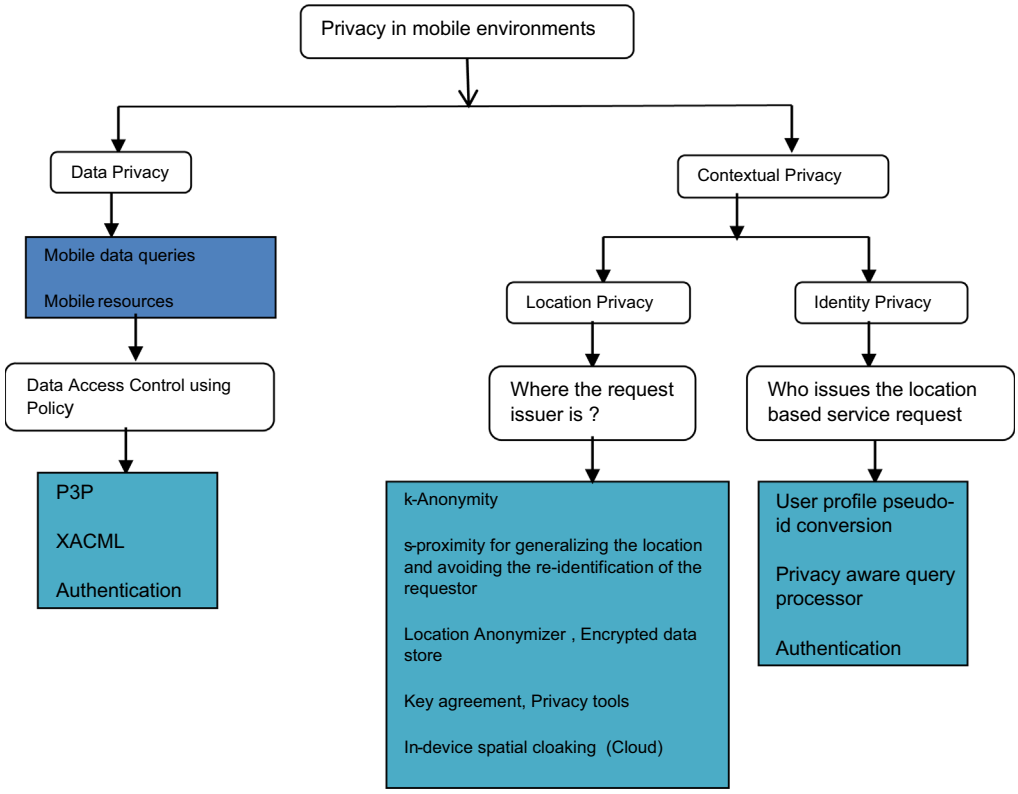

Fig. 1. Classification of privacy preservation mechanisms in mobile environments.

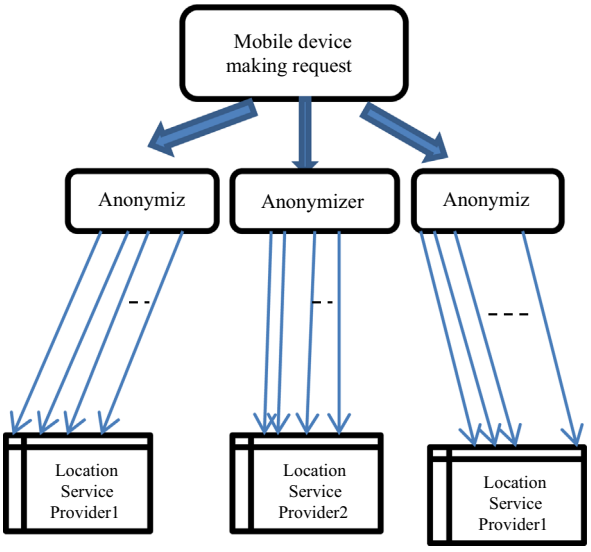

Fig. 2. *k*-anonymity model.

based service providers. This can be achieved by a number of different mechanisms described later on in Sections 15–18. To briefly name the mechanisms here as shown in the architecture diagram above, let us start with *k*-anonymity. In this method, user's location information is updated with pseudo-IDs and then the generalized location information is sent to the location based service provider. Due to some groups being created that fail to provide overall anonymity, another mechanism called *s*-proximity has been implemented (Chowdhury et al., 2009). This mechanism creates a larger number of anonymous user profiles to ensure that the location based service provider cannot identify the location of the requestor. Another location privacy mechanism that is described in this paper is Casper (Mokbel et al., 2006). Casper is a combination of location anonymizer and privacy aware query processor. Few other mechanisms like the encrypted data store (Puttaswamy and Zhao, 2010), key agreement (Loukas et al., 2010), privacy tools (Shokri et al., 2010), In-device spatial cloaking assisted by cloud (Song and Sean, 2010) are also part of the location privacy and are described in detail in the future sections.

Contextual privacy has another classification namely Identity Privacy. Identity privacy mainly talks about the user/mobile server requestor who issues the requests. In order to preserve the identity of the user who issues the requests, a number of mechanisms have been explored. They are mainly user profile pseudo-identifier conversion, privacy aware query processor and authentication based methods. Each one of them is detailed in further sections.

5. Profile anonymization model

Preserving privacy using anonymization has been discussed in a number of research papers (Chowdhury et al., 2009; Liang and Wei, 2008; Riboni et al., 2009; Deivanai et al., 2011). The authors in Shin et al. (2008) have looked at the *k*-anonymity in order to generalize the location. The user of a mobile device usually requests information for two main types of resources namely static resources and mobile resources. In case of static resources, pseudo-identifiers are sent and the location is anonymized. In the case of mobile resources, IDs are updated with pseudo-ids and then the generalized location and profile are sent back to the requestor. Fig. 2 shows the representation of this *k*-anonymity model.

It shows how the mobile device makes a request to one of the location service providers asking for a location-based service. The anonymizer and the location information pick this up and the mobile user information is anonymized and is then transmitted to

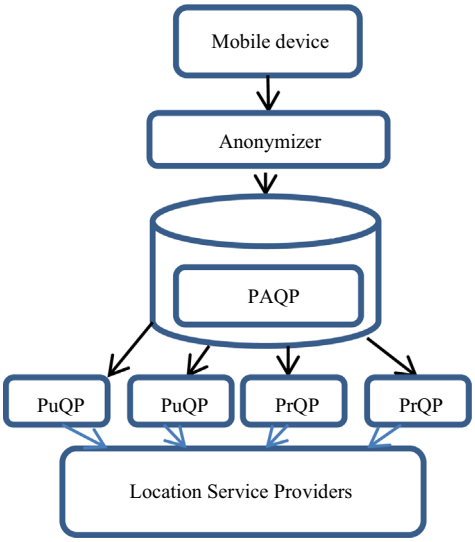

Fig. 3. Casper model.

the location service provider. In this way the user and location information are hidden.

Although the profile anonymization model works well using the *k*-anonymity, there have been a number of attacks that can be performed on the *k*-anonymity model that has led to the identification of the query issuer in the location based services. To overcome some of the shortcomings in the current *k*-anonymity model the *s*-proximity model was proposed. The next section discusses the advantages of the *s*-proximity compared to *k*-anonymity model.

6. Identity inference protection using *s*-proximity in location based services

The *k*-anonymity model as described in the previous section tries to hide the location of the query issuer who tried to request for location based information from the Location Service Provider (LSP). Chowdhury et al. (2009) shows that the *k*-anonymity is not enough as it can be easily prone to attacks thus resulting in the re-identification of the query requester. Two main attacks that have been depicted in the paper are heterogeneity attack and conformity attacks. *k*-anonymity can create groups that fail to provide the overall anonymity due to lack of sufficient match among members with respect to some sensitive user attribute. The communication between the query requester and the LSP is as follows: initially, the user sends a location-based query to the Location Anonymizer (LA), which then replaces the exact location with a Cloaked Region (CR). It is then passed on to the LSP. The attacks prove that in this process, by some combined work by the LSP's or an LSP can individually break down the anonymity set and prove the identification of the specific query requester in cases where the query is specific or not too generic. Hence Chowdhury et al. (2009) come up with a solution that generalizes the query and hence makes it difficult for the LSP to identify the actual query requester. This is achieved in the *s*-proximity model. The paper suggests that both *k*-anonymity and *s*-proximity are needed to anonymize the query requester's identity in a location-based service. In the *s*-proximity model, the LA is replaced by context aware LA with further modules such as query generalization, query analyzer and partitioning agent. With the detailed implementation of these modules privacy of the user is preserved and hence the privacy preservation is achieved in allocation-based environments.

7. Casper: query processing without compromising privacy

The method addresses the issue of user having to give away the location information while requesting for any location-based services through a location based database server. Casper involves two main components namely, location anonymizer and privacy aware query processor. The paper Mokbel et al. (2006) describes in detail how exactly the two main components perform with regard to the four novel areas of scalability, quality, efficiency and flexibility. Casper functions mainly in the following manner. When the mobile user sends the location information along with the query request for a particular location based service, the location anonymizer picks it up and blurs the location information to a spatial region along with the query and passes it to the location based database server. The privacy aware query processor is built into the location based database server and it looks at the request and returns a set of answers that matches the mobile users query. The architecture diagram shows the mobile device making a request to the location based service provider. This is passed through the anonymizer and into the location based database server. The anonymizer does its task and the privacy aware query processor performs its function and the most relevant out of the four data and query would be passed on to the location based service providers (Fig. 3).

The authors (Mokbel et al., 2006) also points out to three novel types of data and query that Casper handles. According to them all the traditional anonymizers can only work on the public query over public data. In Mokbel et al. (2006), the authors propose three novel areas of transactions namely, private query over public data, public query over private data and private query over private data. A detailed analysis of the three methods is shown and the authors assess its performance and scalability.

Casper functionality with private query over public data, public query over private data, and private query over private data can be shown as:

$$\left(q_{pr},\ d_{pu}\right)\ \left(q_{pu}, d_{pr}\right),\ \left(q_{pr}, d_{pr}\right)$$

(M_{lr}) = mobile request and location
A = Anonymization
(C_{lr}) = Cloaking region

$$A(M_{lr}(Q, D))$$

$$Q, D = \left(q_{pr},\ d_{pu}\right)\ \left(q_{pu}, d_{pr}\right),\ \left(q_{pr}, d_{pr}\right)\left(q_{pu}, d_{pu}\right)$$

$$A(M_{lr}(Q, D)) = \{C_{lr}, (Q, D)\}$$

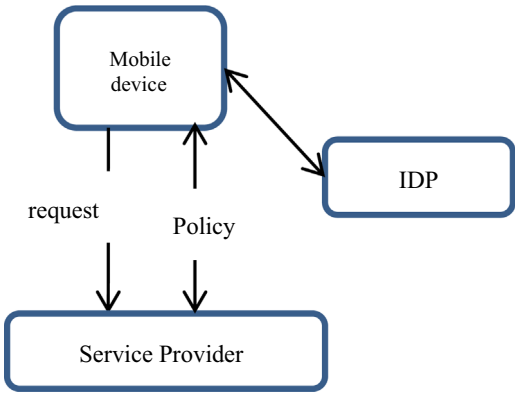

Fig. 4. P3P model for data access control.

$$\{C_{lr},(Q,D)\} = \{C_{lr},(q_{pr},\ d_{pu})\}\{C_{lr},(q_{pu},\ d_{pr})\}$$

$$\{C_{lr},(q_{pr},\ d_{pr})\}\ \{C_{lr},(q_{pu},\ d_{pu})\}$$

According to the authors, by using the Casper's novel solution, the location information will never be compromised. They also address another level of anonymizer called the adaptive location anonymizer, which works, similar to the original location anonymizer with some differences. Details can be found in Mokbel et al. (2006).

8. P3P policy for data access control

Platform for Privacy Preferences (P3P) is a policy that is normally used in websites to negotiate before displaying any webpage to the requester. Arunkumar et al. (2010) have extended this to be used in mobile and ubiquitous environments. The solution proposes to extend P3P policy for controlling the data access in the mobile device. Modifying the P3P policy and using it in the security capsule of a mobile handset achieved this. Security capsule is a software application used in mobile devices (Arunkumar et al., 2010) and it implements security services to protect sensitive data in transit and storage.

A detailed trust establishment mechanism between the security capsule, identity provider and service provider can be found in our previous publication (Weerasinghe et al., 2008). Mobile devices contact the service providers for various services and hence the transaction between the mobile device and service provider involves transfer of sensitive information. The service provider can publish the P3P policy in the Web Service and request the mobile client for the user preferences. With the usage of P3P in the mobile device, the access to the data is controlled including the user preferences and identity mapping. It is also shown in the paper (Arunkumar et al., 2010) that the Service Provider data will always be encrypted and successfully decrypting the data at the mobile end would be a challenge. Hence using P3P policy extension together with encryption and decryption the data access control is maintained (Fig. 4).

There are some extensions that need to be performed on the P3P policy in order to make it work for mobile environments. The process of achieving the privacy and hence data access control can be briefed here as follows: The mobile will first request the sensitive information from the Service Provider and the Service Provider will send an encrypted format of the data to the mobile device. The primary challenge is to provide controlled and appropriate data access control to the right user. This is based on the real time key that is received from the service provider. The service provider sends the data/information requested by the mobile device in an encrypted format. The real time key is used to encrypt the data and the mobile device requires this real time key to access the data. In order to receive the real time key, the mobile client needs to first provide the appropriate user preferences based on the P3P policy of the service provider. The mobile device needs to decrypt the data in order to read the confidential information or in order to access particular information that is sensitive. The mobile client then requests the real time key from the Service Provider. The Service Provider uses this real time key in order to encrypt the sensitive information. In response to this, the Service Provider sends the challenge request with its P3P policy. The security capsule in the mobile device responds to this with the challenge response and P3P user preferences. On the mobile side, Service Provider's policy file is parsed and the identity information that is needed from the mobile device is retrieved. This identity known to the device is then hashed and sent to Service Provider. In the Service Provider side the hashing is carried out and the result is used as the key to encrypt the real time key. Similar method is adapted on the mobile side and the real time key is retrieved. This leads to decrypting the sensitive information. The whole process ensures that the person with the correct access rights is the one who will receive the information.

9. XACML policy in mobile environment

XACML (eXtensible Access Control Markup Language) is a simple, flexible way to express and enforce access control policies in a variety of environments, using a single language. The XACML language in effect protects content from unauthorized use in enterprise data exchanges. XACML is mainly derived around and written in, XML, which is understood in most global environments. OASIS, which drives the development, convergence, and adoption of e-business standards, has ratified XACML. XACML gives an extensive and powerful set of features to the developers. XACML is used to verify the data access control (Arunkumar et al., 2010) in mobile environments. The paper (Arunkumar et al., 2010) talks about XACML and its two main components PDP and PEP. Policy Enforcement Point (PEP) protects the resource when a request is made and sends it to Policy Decision Point (PDP), it then looks at the request and makes the decision based on the access permissions.

The process involved in XACML policy for mobile environment can be briefed as follows. In response to the initial request, the service provider will send a challenge request and a request created by PEP for XACML policy from the mobile device. The mobile client will send the XACML policy with the relevant details in it. Web Services will then pass the request through the PDP, which will look at the request and decide whether the request is eligible to be granted access to the information. Based on the decision made by the PDP, Web Services encrypts the real time key and sends it as a response to the mobile device. The key is then decrypted in the mobile device and the original information is retrieved.

10. Encrypted data store to preserve privacy

Location based social applications (LBSA) are used considerably in today's smartphones. Smartphones using these applications send location information to untrusted third party servers. In Puttaswamy and Zhao (2010) the authors argue that the LBSAs should adapt an approach where the untrusted third-party servers

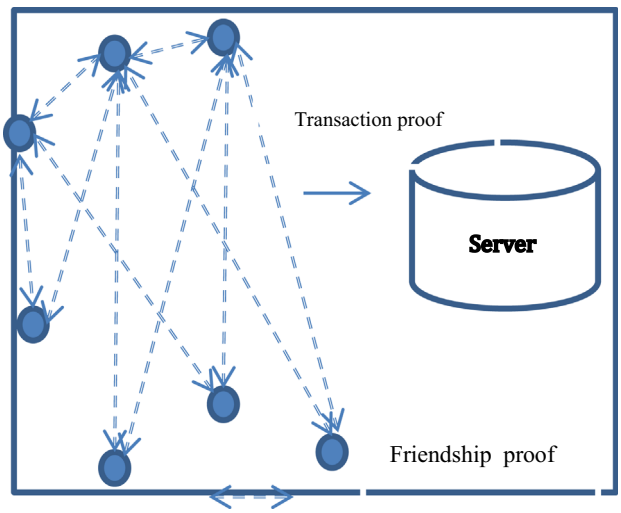

Fig. 5. Encrypted data store model.

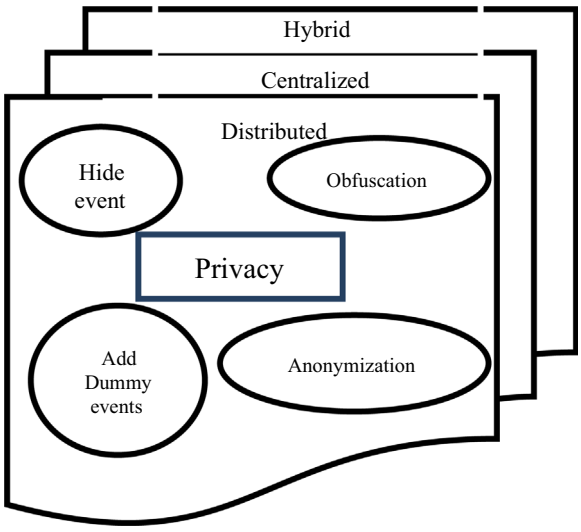

Fig. 6. Location privacy with privacy tools.

are treated simply as encrypted data stores, and the application functionality be moved to the client devices. The location coordinates are encrypted, when shared, and can be decrypted only by the users that the data is intended for. This approach significantly improves user location privacy. The authors also argue that this approach not only improves privacy, but also is flexible enough to support a wide variety of location-based applications used today.

Location information can be easily accessed by the third party servers and hence can be passed on to other sources due to various reasons as mentioned in Puttaswamy and Zhao (2010).

In Puttaswamy and Zhao (2010), the authors propose a design for building LBSAs that provides a low-cost, practical, and deployable alternative to existing design while providing strong user location privacy. The key insight behind this design is to treat the server as a simple encrypted data store, and move the application functionality to the client's smartphone. All the location information shared is encrypted and the lack of plain location information on the storage server improves user privacy. This approach easily works on today's smartphones because the servers running LBSAs today provide their service by running simple operations such as certain database or hash table lookups, performing simple computations on the location data, and sending the results to be displayed on the clients terminals. For example, in a nearby restaurant review application, the server takes the user location, finds restaurants that are in the vicinity of the user's location, queries the reviews of these restaurants, and sends the results back to the users for display. In the proposed approach, the data storage and lookup operations happen on encrypted data but still remain on the storage server. The clients receive the encrypted results, decrypt and display the results to the users. The clients only incur an additional cost of decrypting the received content, and perform simple calculations on the decrypted data.

Figure 5 shows that friends exchange friendship proofs and store them in their devices and then users generate and store the transaction proofs in the server and their friends later on retrieve this.

By using lightweight cryptographic schemes such as encryption, decryption with real time keys, the authors claim that they can easily move the functionality to the smartphones and provide services while preserving privacy. The paper discusses two proofs namely, friendship proof and transaction proof.

Friendship proofs cryptographically attest the social connection (or friendship) between two users, and similarly, transaction proofs cryptographically attest certain data generated by a user. Using these proofs, any user in the network can verify if it is a friend, and if so decrypt the data generated. But no other user other than a friend will be able to see the contents. Finally, the interface exposed by the storage server is narrow enough that one can reason about the privacy guarantees, and yet they are flexible enough to build several LBSAs. As a result, a single storage server can support many different LBSAs.

11. Unified framework for location privacy

According to paper (Shokri et al., 2010) there are three entities that play a role in preserving location privacy: users, applications, and privacy tools. Each entity controls the amount of shared information and thus affects user privacy. Users and applications might intentionally (e.g., by being cautious about sharing unnecessary information) or unintentionally (e.g., by sharing incorrect information) reduce the amount of information revealed. Privacy policies influence the way applications can share information with different entities, and they are applied to the application based on the users' decisions. Various privacy tools (Shokri et al., 2010), also, use sophisticated algorithms to guarantee users' privacy. In order to capture the effect of the three entities in preserving location privacy of users, in Shokri et al. (2010), they abstract the entities and model a location-privacy preserving mechanism as a single unit that separates actual events of the users and the adversary. Paper (Shokri et al., 2010) defines a location-privacy preserving mechanism as a transformation function that modifies the users' actual events before they can become observable by any observer. The paper discusses the privacy tools in detail (Fig. 6).

Privacy tools work in three architectures: (i) *Distributed* (user-side): They can work in a distributed way by being implemented on individual mobile devices, where each device transforms its events and modifies what an observer can see about the user's spatio-temporal state. This can be done either with the help of information that a device gets from other devices or exclusively with the information that the user has. (ii) *Centralized* (server-side): They can work in a centralized manner by using a trusted central server that acts as a privacy preserving proxy and modifies users' messages (correspond to events in our model) before being observable by an untrusted entity. (iii) *Hybrid*: They can be a hybrid of both distributed and centralized architectures.

The four main functions in the location privacy preserving mechanism include hiding events, adding dummy events, obfuscation and anonymization.

12. Authentication and key agreement for location privacy

Loukas et al. (2010) discussed in their paper about mobile instant locator with chatting capability along with preserving privacy and security. Mobile instant locator with chatting (MILC) was developed for usage within a closed community and hence worked very well in the University scenario described in the paper. The paper (Loukas et al., 2010) also highlights that with its popularity grew its demand and since it also incorporated privacy preserving techniques it was very attractive to other communities too. MILC works towards making the communication confidential and maintaining the privacy of the user. According to Loukas et al. (2010) the MILC server is developed in Java. The client server communications are handled using the RSA 1024 bit asymmetric keys. Client gets successfully authenticated with the server and from then onwards every communication between the two ends is secured by using a symmetric session key created at the server end. The paper (Loukas et al., 2010) proves that supporting pseudonymity and location privacy can preserve the end-user

privacy. The option of presenting or disclosing the location is left to the choice of the user. If the user decides not to disclose the location, user's privacy is maintained. Pseudonymity is provided per session. When the user connects to the MILC server is offered with the option of choosing a different pseudonym for the current session. In the paper (Loukas et al., 2010), authors have also compared MILC with three other applications (⟨http://www.androidapps.com/t/gfindster⟩), (⟨http://im-easy.com/⟩), (⟨http://www.buddymob.com/⟩) and show how comparison based on security requirements. The comparison is based on the following six basic criteria; mutual Authentication, confidentiality, integrity, pseudonymity, resistance to DoS (Denial of Service) caused by insiders and location privacy. The comparative view of all the applications considering the above mentioned seven basic criteria show that the applications support user authentication and pseudonymity. MILC additionally provides mutual client–server authentication. Moreover, the pseudonym of a MILC user cannot be associated with the permanent identity in any way. Excluding MILC, BuddyMob is the only one supporting location privacy, but this applies for guest users only.

13. In-device spatial cloaking assisted by Cloud

A number of privacy mechanisms proposed mostly deal with single point of service and when there is a single point of service, things are bound to go wrong somehow somewhere. Song and Sean (2010) talk about the cloud services available that makes it so much more versatile in terms of the services being available in the cloud. The authors describe how the location based services that are requested by the mobile device are delivered to them by means of using spatial cloaking that is assisted by cloud capabilities. There are clients in the mobile devices that would be responsible for generating the cloaking region. The main difference of the In-device spatial cloaking solution in comparison to the Casper solution is that here it is the device generating the cloaked region and hence the paper strongly portrays that using the in-device cloaking privacy can be preserved. The in-device spatial cloaking solution involves a location trusted server, which takes the information from the mobile device strips that information and carries only the spatial cloaked information and the service request and passes it on to the service provider.

14. Open problems

The in-device spatial cloaking solution involves a location-trusted server, which takes the location information from the mobile device strips that information and carries only the spatial cloaked information and the service request and passes it on to the service provider.

The solution needs the grid structure to be kept inside the memory of the mobile device and this grid structure needs to be up-to-date with the device. The paper proposes a top down cloaking algorithm in comparison to the bottom up approach of Casper model.

There are a number of challenges and loopholes in each of the privacy preserving techniques described in this paper. The individual papers highlight the drawbacks or the challenges in the proposed solutions. The *k*-anonymity solution described in the paper has a number of issues associated with it. With the limited number of profiles created, it becomes easy for the location service providers to easily track down the actual requestor and further identify the location of the requestor. This problem is clearly explained with an example in Chowdhury et al. (2009) which further proposes *s*-proximity. The *s*-proximity solution overcomes the problem of location service provider identifying the requestor and location when there are a certain large number of profiles. However, if this large number of profiles is not large enough, then the same problem as *k*-anonymity will start to appear. Casper model that is spoken about in this paper has a location based database server and an anonymizer, which takes care of different types of data over different queries. It is important to note that the database server is a single service and hence can be prone to a number of attacks. Hence Casper solution needs to be further enhanced. With the P3P and XACML policy extension mechanisms described in this paper, there are obvious limitations of P3P and XACML. Hence negotiation between the server and the mobile device needs to be implemented using a policy language that ensures compatibility on the server side and within the mobile applications requirement. The focus for future work in these two mechanisms will be to come up with a novel policy language for the enforcement and policy negotiation between the Web Service and the mobile device before transferring any sensitive information to the device. In the encrypted data store mechanism described in this paper, the challenge is to extend the solution with new mechanisms for users to securely discover the keys used to encrypt the data on the server, without revealing the key to the server itself. This is an area, which can be further explored by the scientific community. The privacy tools mechanism focuses on the location unified framework used for preserving the location. With the unified framework, there are certain challenges due to the emerging threats related to time and location. The solution has some problems with the accuracy of the location privacy metrics. This helps us to focus on future research in location privacy and its elements including anonymity. MILC technique is used in a small scale in a University environment and hence this solution works well in a closed community. However, when the solution is proposed to be used across a wider community, the risks of security and privacy are high. Hence this solution needs to be looked into much more detail in terms of the location privacy and scalability. The specification of user being able to decide whether to give away the information of his location seems to be much more complicated in an enterprise setup. This section has been mainly written to summarize the challenges and to list down the open issues. The In-device spatial cloaking module that is assisted by the cloud solution is a good start to a solution based on cloud services. This solution talks about the location trusted server being in the cloud and the mobile device itself generating the spatial cloaking with the help of the up-to-date grid structure. With this solution, it would again be different to accurately make the grid capabilities to up to date.

15. What, how and where of location privacy?

There are 3 main questions to be answered when we consider location privacy. What needs to be preserved and how privacy can

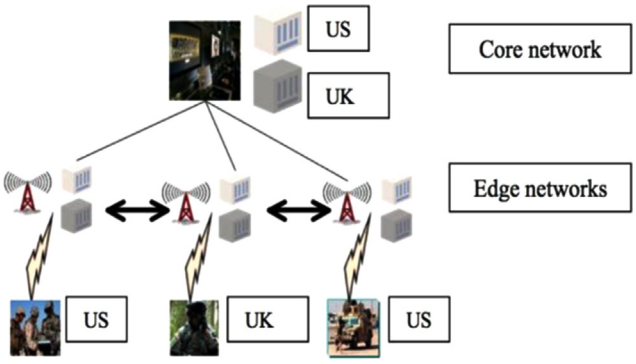

Fig. 7. A tactical network scenario enabling efficient computations over dynamic networks.

be preserved – these two questions are the standard ones that all researchers have been addressed. The review work that has been described so far in the paper all relates to what and how.

But the one question that has not been addressed before is *"Where to enforce Privacy?"*

Our attempt at exploring ways to find out where exactly privacy needs to be enforced has led us to conclude that an edge based solution is where privacy should be enforced.

This section focuses on Device vs. Edge based implementation and the tradeoffs in them. This section quantifies the tradeoffs and proves that edge based solution is the better solution for enforcing security. Our results show that while device-based solutions do not require trust in the edge location server, they either suffer from high false positive rate (about 25% probability of not meeting the desired security requirement) or low utility (about 600 meters higher error in obfuscated location data).

15.1. Solution at the core

The core is the centralized network and hence has a lot of bandwidth and can maintain huge repository of information. It also has a lot of computational power allowing it to process complex solutions. It is important to note that it takes longer time for transactions to work between the device and the core. This is a major drawback to the location-based solution, as decisions need to be made rapidly else will lead to delays in the decisions to be taken and hence weakens the system. Solution at the core will retain same false positive and false negative and will have a very high latency.

15.2. Solution on the device

The delays caused due to the solution being placed at the core of the network gave rise to the new wave of solutions that were placed on the device. It is important to notice that the device does not have a lot of flexibility, bandwidth, and computation power. Besides any of these, the device does not have visibility of the other devices in the network. Hence any kind of computations performed by the device will not be leading to accurate results. It could very well lead to misleading answers to the user's request.

This leads us to the new methodology that we introduce in this paper called the solution at the edge of the network.

15.3. Solution at the edge

The edge of the network is closer to the device and is an intermediate channel between the device and the core of the network. The edge has visibility of all the other users in the network and the edge can perform computations faster and provide with results spontaneously to the device. The advantage of having the solution at the edge is that edge will have information about other people and hence solution will have lower false positive and lower false negative. The only catch with this solution is that trust with the edge is needed. The edge will have the raw obfuscated data or slightly obfuscated location data. Latency with this solution is higher than device based solution and is lower than the solution at the core. This helps the device user make decisions on the location based service requests that one has. Hence this solution is the best solution compared to the three solutions explained.

16. Mobile microcloud

Introducing mobile micro-cloud in this paper will help in understanding the placement of the solution. Mobile micro-cloud (Warr and Zafer, 2013) envisions that applications (or computing tasks) will be deployed in a mobile micro-cloud, a logical network composed of two components, the core (e.g., the command and control center) with access to large quantities of static (and possibly stale) information and the edge (e.g., the forward operating base) with access to smaller quantities of more real-time and dynamic data. The edge and core are separated by dynamic and performance constrained networks with a many-to-one relationship between the core and the edge. It is also possible for edge nodes to communicate with each other. Further, the (edge and core) nodes can belong to different coalition partners, raising the question of security and operational policies for handling of data and computation.

Figure 7 illustrates a typical architecture of the mobile micro-cloud in the army coalition context. The benefits of embedding storage and computation into such a micro-cloud tactical network are two folds: (I) effective provisioning for diverse information requirements the micro-cloud supports users with different latency requirements and access rights and (ii) effective information exchange in a constrained environment. Complete shuffling of information is impractical in a tactical network and the micro-cloud reduces congestion by providing computation at the edge.

Privacy solutions could also work in a cloud based environment but a Microcloud based solution will have low latency in comparison to the cloud based solution; however, if the cloud based solution can avoid low latency and can be placed somewhere near the device, the solution would still be technically correct.

17. Security metrics

This section presents an empirical evaluation of the proposed location information flow control solution. Table 1 shows a summary of the datasets used for evaluation. Three of the datasets Shanghai, San Francisco and Stockholm are taxicab traces obtained from the respective cities. The fourth (Cellular) is a user location trace and URL accesses obtained from a cellular network. The fifth (Watson) is an enterprise dataset obtained from WiFi location traces and URL accesses.

Table 1
Summary of datasets.

Characteristic	Shanghai	San Francisco	Stockholm	Cellular	Watson
Sampling rate	2/min	12/min	1/min	>400/day	All web access
Number of entities	$\sim$10,000	$\sim$500	$\sim$200	$\sim$16,000	$\sim$1200
Source type	GPS	GPS	GPS	Cellular base station association	WiFi
Privacy	None	None	No sampling when taxi hired	Course grained samples	None
Timeline	1 month	1 month	1 month	1 month	1 month
Total number of trips	1,335,360	26,767	570,690	55,200	–
Total number of web accesses	–	–	–	12.4 M	5.6 M

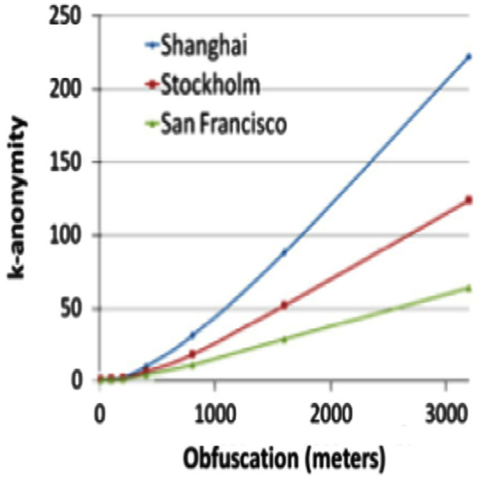

Fig. 8. 7–10 am.

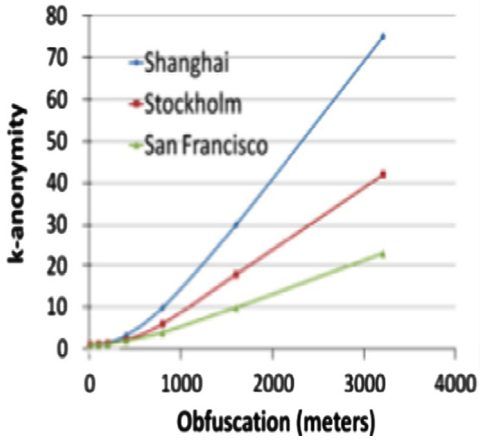

Fig. 9. 10 am–4 pm.

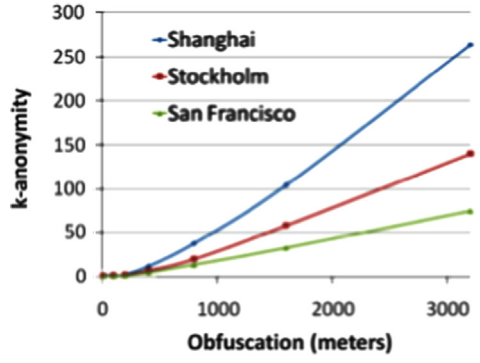

Fig. 10. 4–7 pm.

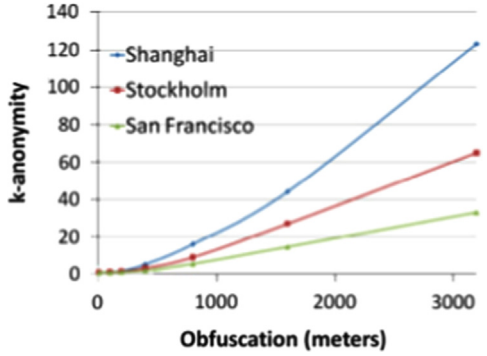

Fig. 11. 7 pm–7 am.

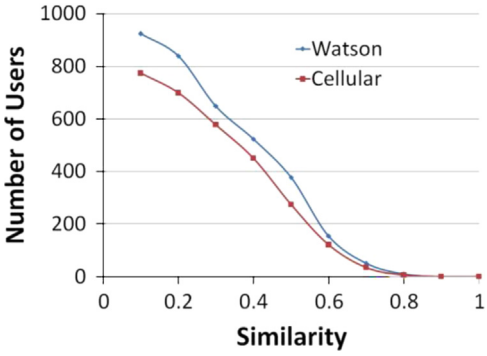

Fig. 12. Similarity of user profiles (based on data accesses).

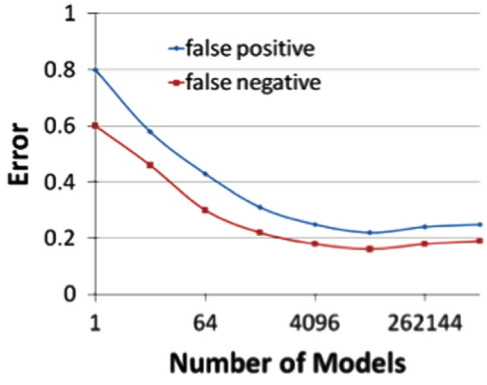

Fig. 13. Shanghai.

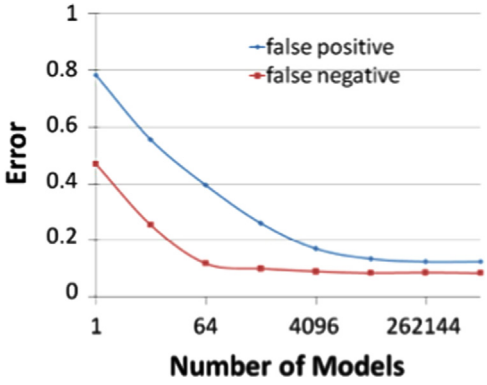

Fig. 14. Stockholm.

In the Shanghai and San Francisco datasets, there are explicit markers that indicate when the taxicab is occupied; in the Stockholm dataset collection of location trace is turned off when the taxicab is occupied (i.e., we only have trajectory information when the taxicab is not occupied). We use these datasets to quantify tradeoffs between the extent of obfuscation and anonymity.

In addition to these datasets, we use coarse-grained mobility data from 16K mobile users obtained from CDRs (Call Detail 2 Records) and from about 1.2K enterprise users obtained from WiFi and web data accesses. While a taxicab's trajectory may be viewed as a mixture of several user trajectories (i.e., multiple passenger trajectories), this dataset captures movement information at the granularity of each user. However, location information is captured is at the level of cellular Base station association, which depending upon urban/rural areas can range from a few 100 m to about 5000 m. From a population of about 11.6 M users, we selected about 16K users that

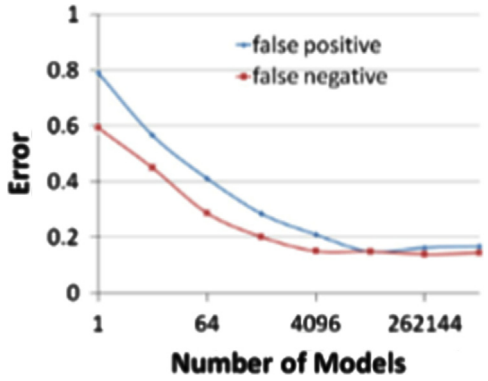

Fig. 15. San Francisco.

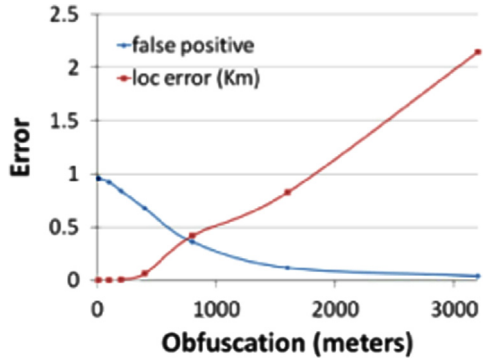

Fig. 16. Shanghai.

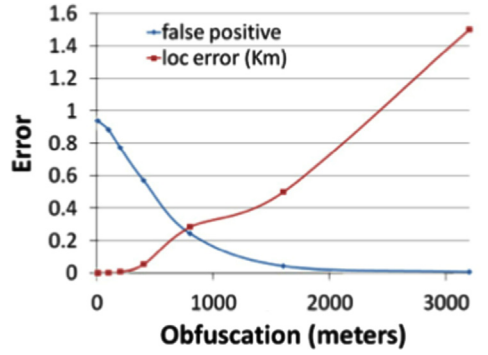

Fig. 17. Stockholm.

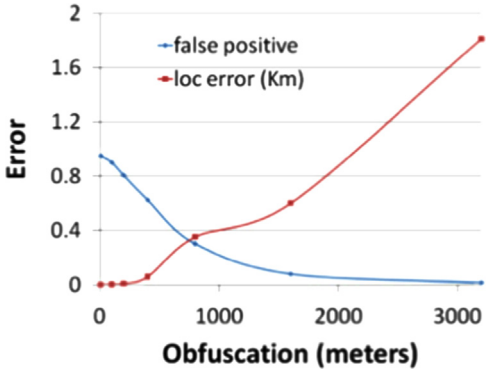

Fig. 18. San Francisco.

had more than 400 CDRs per day (i.e., >400 location samples and data accesses per day). While we use the taxicab dataset to analyze fine-grained trajectories (each corresponding to one trip), we use the cellular and enterprise dataset to analyze mobility across multiple trips undertaken by a single user.

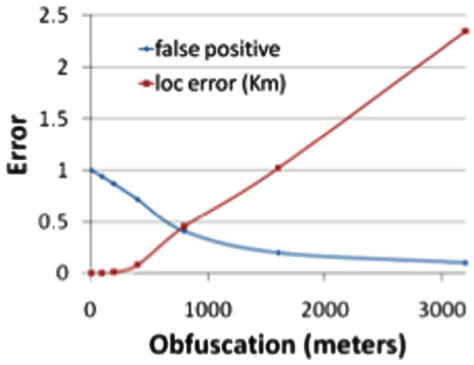

Fig. 19. Cellular Thr 0.0.

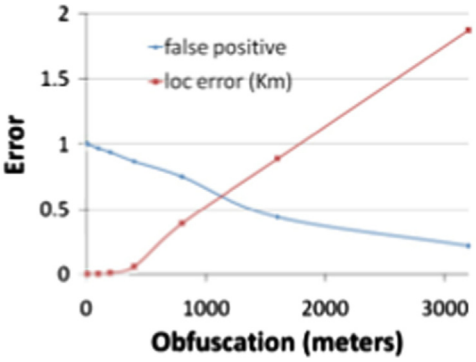

Fig. 20. Cellular Sim Thr 0.7.

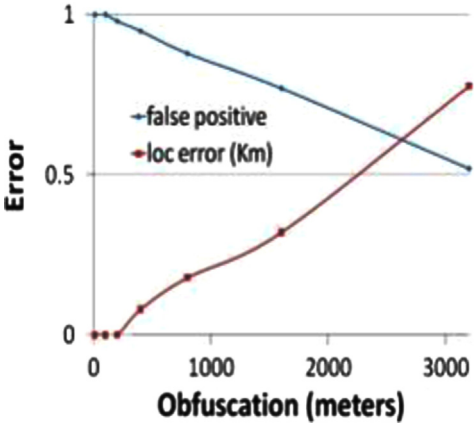

Fig. 21. Cellular Sim Thr 0.9.

Figures 8–11 show the average anonymity as the extent of obfuscation is varied for times 7–10 am, 10 am–4 pm, 4–7 pm and 7 pm–7 am respectively. As the extent of obfuscation is increased so does the extent of anonymity; further anonymity is generally higher during busy hours in the morning and the evening because several mobile users are active within a small spatial extent. The key challenge in practice is that these measures of anonymities are averages over the respective dataset. Hence, given a user location at a point in date and time, the challenge is to identify the amount of obfuscation required to achieve a desired level of anonymity.

Figure 12 shows the number of users on the y-axis and similarity on x-axis. A point (x, y) in the figure indicates that there are at least y users whose profiles have a similarity of at least x with a randomly selected user. Similarity between user profiles is

computed using a cosine distance on the set of URLs (web pages) accessed by a user with that of another user.

Figures 13–15 show the complexity of a device-based model and false positive and false negative rates in enforcing the desired level of anonymity. A choice of obfuscation k is said to result in a false positive if it results in cloaking $< k$ users.

And in a false negative if it results in cloaking $\geq k$ users. A false negative is an indicator of over obfuscation, which would in turn affect the utility of the obfuscated data; while a false positive is in direct violation of the k-anonymity security requirement. In order to determine the level of obfuscation we analyze historical data using decision tree based machine learning algorithms – parameterized by location (typically encoded as latitude/longitude boxes) and timestamps (typically time of day and week). We tradeoff model complexity (i.e., number of nodes in the decision tree) with accuracy (i.e., being able to predict the desired level of obfuscation). We observe that increasing model complexity beyond a desired level increases the error primarily due to over fitting. We also noticed that in most cases the false positive and false negative rate of an optimal device-based algorithm (with large model complexity) varies between 0.12 and 0.25 for our datasets. This captures the extent of sub-optimality in device-based solutions in comparison with an edge-based solution.

Figures 16–18 show the false positive rate (i.e., the odds of not meeting the desired level of anonymity) and location error. Location error is only computed when the choice of obfuscation meets the desired level of anonymity. If the choice of obfuscation meets the desired level of anonymity and nothing more than location error is zero. Otherwise, location error is computed as the difference between the extent of obfuscation chosen and the optimal obfuscation needed to achieve the desired level of anonymity.

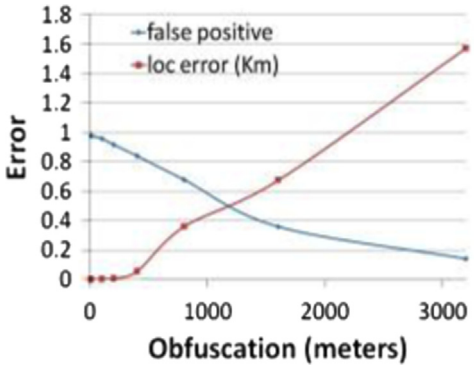

Fig. 22. Shanghai Sim Thr 0.7.

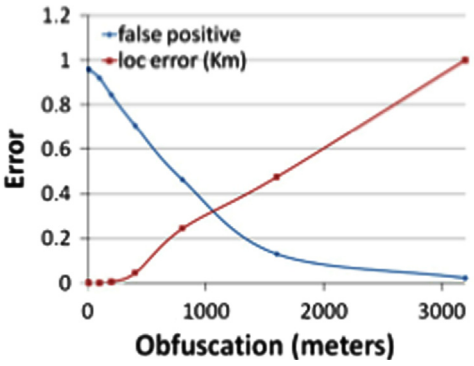

Fig. 23. Stockholm Sim Thr 0.7.

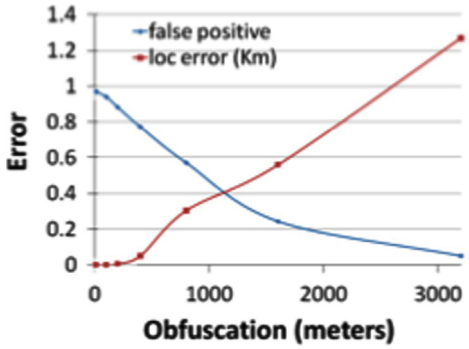

Fig. 24. San Francisco Sim Thr 0.7.

Figures 19–21 show the false positive rate (i.e., the probability of not meeting the desired level of anonymity) and location error with and without consideration to user similarity.

For this experiment the desired level of anonymity $k=16$ and the desired level of user similarity is 0.0 (first case that ignores user profiles), 0.7 (in the second case) and 0.9 (in the third case). For instance when user similarity threshold is 0.7, amongst the set of users that are within the extent of obfuscation only those users whose profiles are at least 70% similar to the given user are considered for quantification of anonymity. This figure shows the additional cost (higher false positive rate and higher location error) that is incurred when enforcing location security based on profile cloning. We observe that when the similarity threshold is low the device-based solution pays a high penalty in terms of location error, while when the threshold is high the device- based solution pays a higher penalty in terms of false positive rate (i.e., the inability to meet the security requirement).

Figures 22–24 show the false positive rate (i.e., the odds of not meeting the desired level of anonymity) and location error while requiring a user similarity threshold of 0.7. Profiles for entities are drawn at random from the Watson dataset with the goal of showcasing tradeoffs between location security and identity/profile based obfuscation. Similar to prior experiments, location error is only computed when the choice of obfuscation meets the desired level of anonymity. If the choice of obfuscation meets the desired level of anonymity and nothing more than location error is zero. Otherwise, location error is computed as the difference between the extent of obfuscation chosen and the optimal obfuscation needed to achieve the desired level of anonymity.

18. Android based implementation

This work has been implemented as an android based system. An application has been implemented in the android device in order to showcase the difference in the two methodologies. The solution at the device and the solution at the edge have been implemented using an example of the London Boris bikes. Boris bikes are the easiest way to hire a cycle, ride it where you like and return it to any docking station. In this implementation, we have shown the means of how the system solution works when the solution is at the edge and when it's at the device. In order to perform the implementation, we have made use of an application in an android device and then have implemented an edge server on a windows server. This server behaves as an edge, which has the visibility to all the devices in the network, and performs computations accordingly. The device based solution shows an android application with the map of London in it indicating the Boris bikes available for hire. Request from the mobile device is shown on the map by indicating the current location of the device. By performing obfuscation on the device, it can be noticed that the

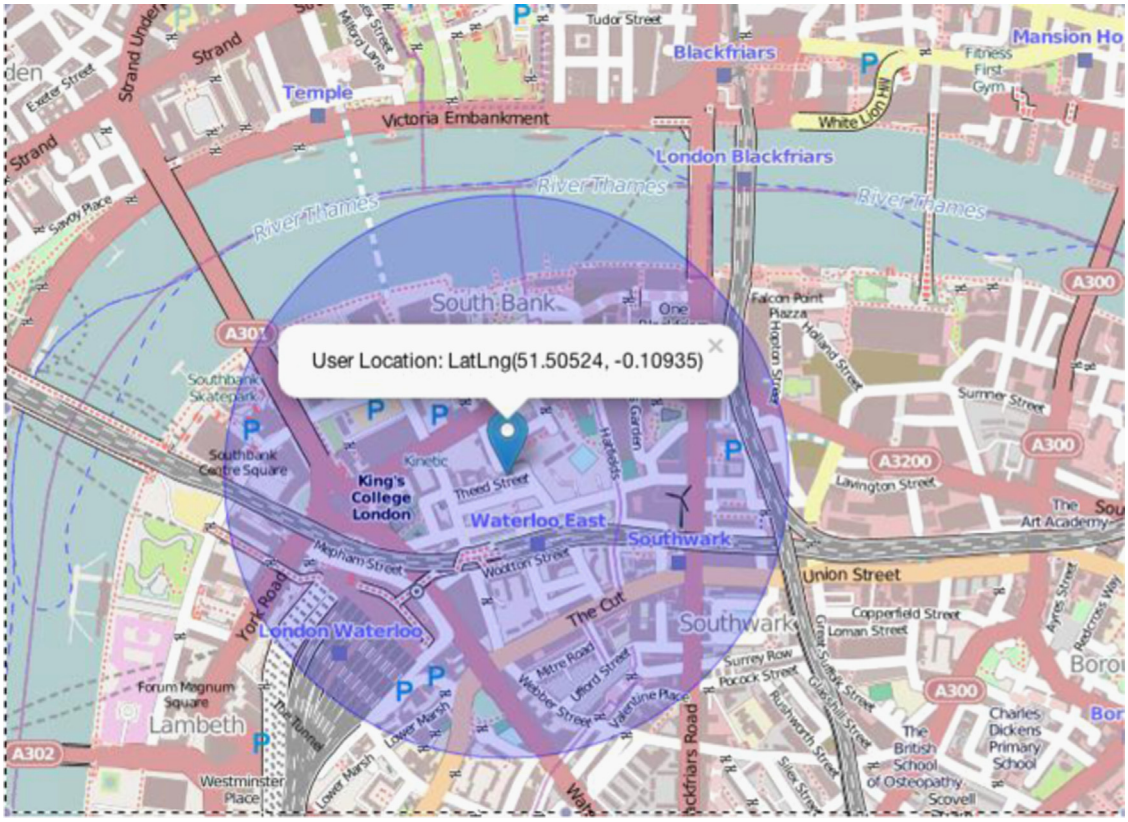

Fig. 25. Device based solution view of the London Thames region.

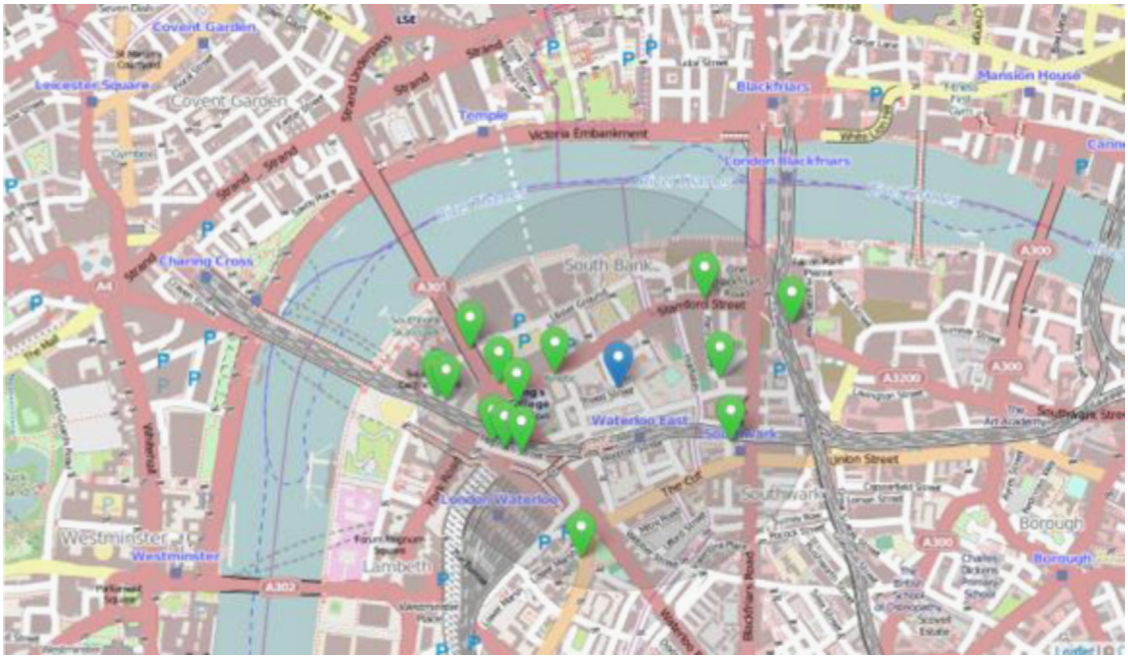

Fig. 26. Search results for the device based solution.

obfuscation is not accurate enough as the device does not have visibility to other devices in the network. When the user then makes a request for the bikes, the responses received are not accurate due to the drawback of inaccurate obfuscation. In the case of solution at the edge, the edge has visibility to all the devices. When the user makes a request asking for the nearest bike hire from the current location, the edge takes care of obfuscating the current location of the device in comparison with the other devices in the network that would have made similar requests. The request is then sent from the obfuscated location and this results in accurate responses for the user requesting the locations of the bikes nearby from his location. Figs. 25–28 show the different stages in the demonstration of the location-based request with the anonymized location and the results of the query. The solution has been implemented using the Eclipse development kit and has been tested with real use case scenarios. Fig. 25 shows the

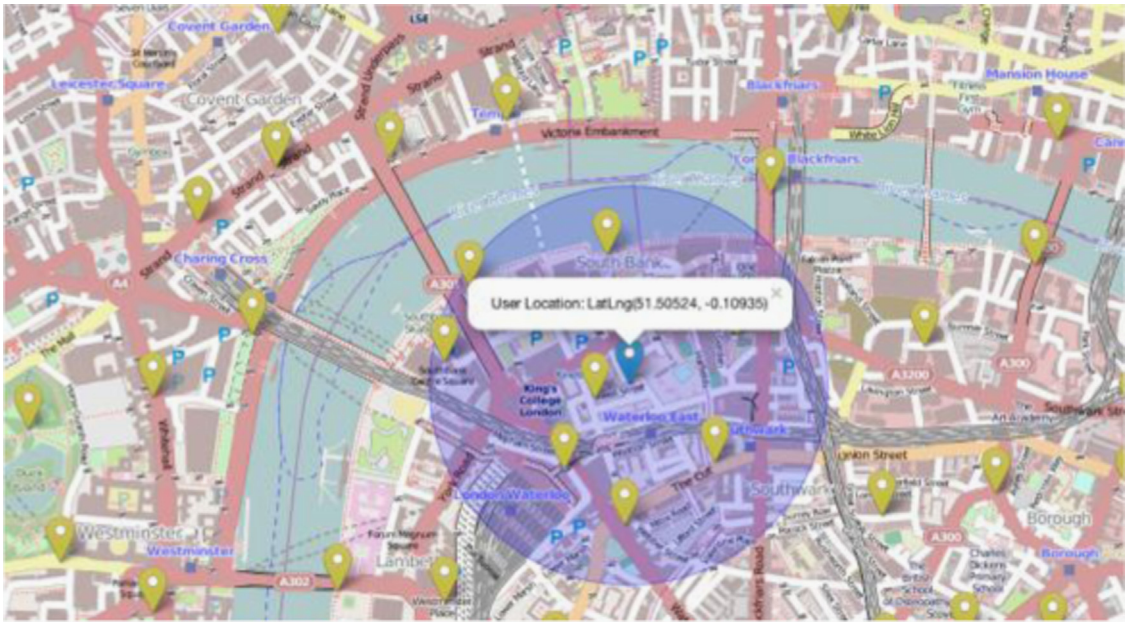

Fig. 27. Devices that are visible to the edge server.

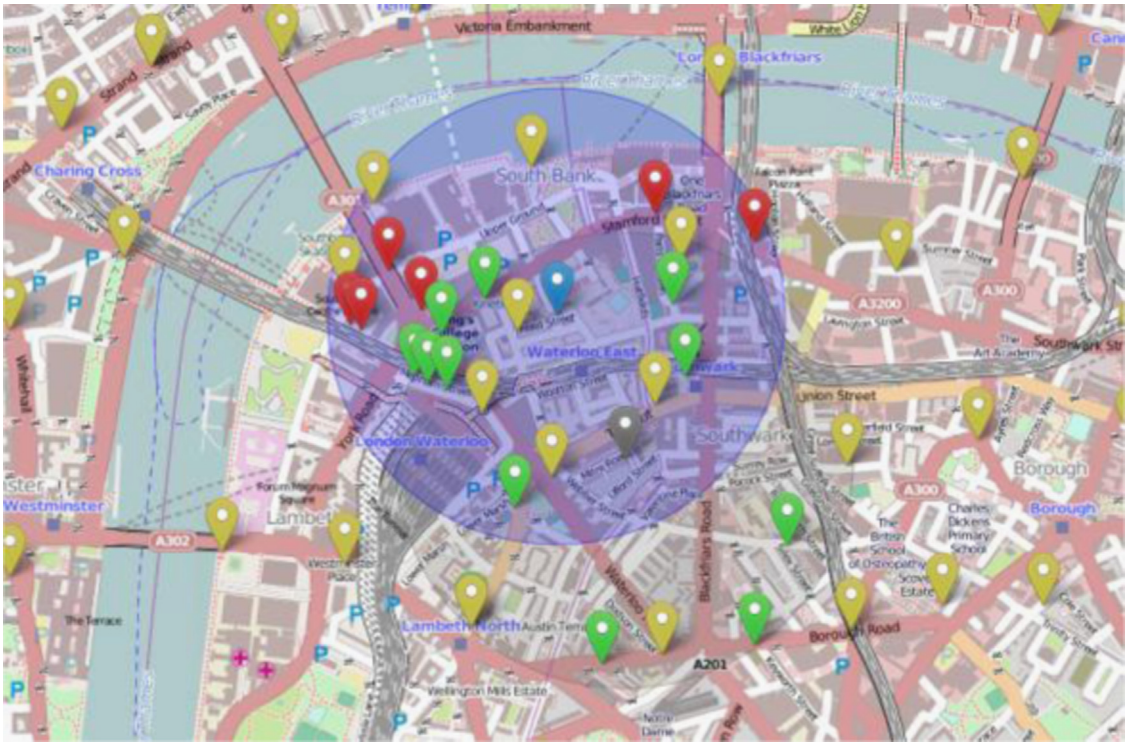

Fig. 28. Query results from true and obfuscated location.

device-based solution where user clicks on a particular point and then checks are done to see if the chosen location has enough obfuscation.

Device level obfuscation cannot be performed, as the device has no visibility to the other devices.

Hence checks are done at the edge server to ensure that the obfuscation is good enough to make a query. Fig. 26 shows the search results for Boris bike using the device-based solution.

Figure 27 shows the view that the edge server would have of all the devices. Since the server can see all the devices, when a device makes a request for the bikes, the server can obfuscate the location based on the other devices in the area.

On searching for the bikes based on the new obfuscated location, the results are displayed in Fig. 28. The comparison of results based on the search from the true location and the obfuscated location is shown using the two circles. This proves

that the edge server functions close enough to the query made directly to the Boris bikes provider without any obfuscation.

19. Conclusion

Preserving user privacy is a very challenging issue in the mobile environments. Today's mobile devices have become much more capable of doing things one would not have imagined 10 years ago. With the location services and the capabilities of the applications in the mobile devices, service providers can personalize any type of service that one asks for, from finding a stolen phone anywhere in the world to providing all the latest information about a new restaurant opened in the neighborhood. The question is how one makes use of the location information of an individual user and how the privacy of the user information is preserved. This has been a question for researches for many years and the problem is getting worse day by day due to the change in privacy policies of major service providers like Google and Facebook who have actively harvested data over the last number of years and are now changing their polices to make use of these harvested data to deliver new personalized services to the customers. There are various solutions and mechanisms that have been provided and prototyped by many research groups and companies over the last several years. However, due to the increasing connectivity between new services and the inter-dependency between the service providers is making the privacy management a challenging hard task for an innocent user of the mobile devices This review paper is an attempt to review all of the existing privacy preserving techniques that has been proposed for the mobile environments and identify some of the flaws in the existing techniques that needs to be overcome to make the mobile a safer and secure platform to transact and communicate in the future.

We have explored both device and edge based enforcement of location security and quantified the gap between optimal device-based enforcement with that of the edge-based enforcement. In particular, we have identified machine-learning algorithms that determine the extent of location obfuscation that is needed to achieve a desired level of anonymity. We have shown that even with good models a device based solution (that is unaware of the instantaneous locations of other entities or their profiles) is largely suboptimal in determining the extent of location obfuscation. Our experiments on various mobility datasets show that device-based solutions either suffer from high false positive rate (about 25% chance of not meeting the desired security requirement) or low utility (about 600 m higher error in obfuscated location data).

Acknowledgment

We sincerely acknowledge the valuable discussions and input from Dakshi Agrawal, Manager, Network Management Research Group, Hawthorne, USA. We sincerely thank Ian Robertson, Senior managing consultant, Security and Privacy, IBM UK Limited, for his valuable comments and his role as industrial research supervisor of the first author.

References

Arunkumar S, Raghavendra A, Weerasinghe1 D, Patel D, Rajarajan M, Policy extension for data access control. In: Proceedings of 6th IEEE workshop on secure network protocols (NPSec), Japan,2010. p. 55–60.

Barkhuus L, Dey AK, Location-based services for mobile telephony: a study of user's privacy concerns. In: Proceedings of the international conference on human-computer interaction, Switzerland, 2003.

Beresford A, Stajano F, Mix zones: user privacy in location-aware services. In: Proceedings of IEEE workshop on pervasive computing and communication security (PerSec), 2004. p. 127–131.

Beresford Alastair R. Location privacy in ubiquitous computing. UCAM-CL-TR-612. University of Cambridge; 2005 ISSN 1476-2986.

Brar A , Kay J, Privacy and security in ubiquitous personalized applications, Technical Report no. 561, ⟨http://www.it.usyd.edu.au/research/tr/tr561.pdf⟩,2004.

BuddyMob 2009. Retrieved from ⟨http://www.buddymob.com/⟩.

Cai L, Machiraju S, Chen H, Defending against sensor-sniffing attacks on mobile phones. In: Proceedings of the 1st ACM workshop on networking, systems, and applications for mobile handhelds (MobiHeld '09), New York, NY, USA, (ACM), 2009. p. 31–36.

GFindster. GFindster review 2008. Retrieved from ⟨http://www.androidapps.com/t/gfindster⟩.

IMEasy. Introduction to Hi AIM 2008. Retrieved from ⟨http://im-easy.com/⟩.

Chowdhury, S. Hasan, Ahamed Sheikh I., Tanviruzzaman.M, A privacy enhancing approach for identity inference protection in location-based services. In: Proceedings of the33rd annual IEEE international computer software and applications conference, 2009.

Deivanai.P, Vedha .J. Jesu Nayahi, V.Kavitha, A Hybrid data anonymization integrated with suppression for preserving privacy in mining multi party data. In: Proceedings of the IEEE-international conference on recent trends in information technology, ICRTIT, 2011.

Enck W, Gilbert P, Chun B-G, Cox LP, Jung J,McDaniel P, et al. TaintDroid: an information-flow tracking system for realtime privacy monitoring on smartphones. In: Proceedings of the 9th USENIX symposium on operating systems design and implementation (OSDI'10), 2010.

Escudero A, Hedenfalk M, Heselius P, Flying freedom, location privacy in mobile internetworking. In: Proceedings of the INET2001, 2001.

Escudero-Pascual A Holleboom T, Fischer-Hubner S, Privacy for location data in mobile networks. In: Proceedings of the nordic security workshop, NORDSEC, 2002.

Fedler R, Schutte J, Kulicke M. On the effectiveness of malware protection on android. Technical report. Berlin: Fraunhofer AISEC; 2013.

Gedik B, Ling L, Location privacy in mobile systems: a personalized anonymization model. In: Proceedings of 25th IEEE international conference on distributed computing systems (ICDCS), 2005. p. 620–629.

Ghinital, G Kalnis, P, Skiadopoulos S , Mobihide: A mobile peer-to-peer system for anonymous location-based queries. In: Proceedings of the international symposium on advances in spatial and temporal databases, SSTD, 2007.

He Q, Wu D, Khosla P. Quest for personal control over mobile location privacy. IEEE Commun Mag 2004;42(5):130–6.

Jamaluddin J, Zotou N, Coulton P, Mobile phone vulnerabilities: a new generation of malware. In: Proceedings of the IEEE international symposium on consumer electronics, 2004. p. 199–202.

Karyda M, Gritzalis S. Privacy and fair information practices in ubiquitous environments, research challenges and future directions. Internet Res 2009;19(2):194–208.

Langheinrich M, A privacy awareness system for ubiquitous computing environments. In: International conference on ubiquitous computing, 2002. p. 237–245.

Lia J, Qi YH, Huang PW, Rong MT, Li SH. Protection of mobile location privacy by using blind signature. J Zhejiang Univ Sci A 2006;7(6):984–9.

Mobiloco - location based services for mobile communities.⟨http://www.mobiloco.de⟩.

Liang Z, Wei R, Efficient k-anonymization for privacy preservation. In: Proceedings of the 12th international conference on computer supported cooperative work in design, 2008. p.737–742.

Liu L, Yan G, Zhang X, Chen S. Virusmeter: preventing your cellphone from spies. In: Kirda D, Jha S, Balzarotti D, editors. RAID, Vol. 5758. Springer; 2009. p. 244–64 of Lecture Notes in Computer Science.

Loukas A, Damopoulos D, Menesidou S A, Skarkala M E, Kambourakis G, Gritzalis S. MILC: A secure and privacy-preserving mobile instant locator with chatting. Springer Science plus Business Media, LLC; 2010.

Minch RP, Privacy issues in location-aware mobile devices. In: Proceedings of the Hawaii international conference on system sciences, 2004. p. 1–10.

Mokbel MF, Chow CY, Aref WG, The new casper: query processing for location services without compromising privacy. In: Proceedings of the VLDB, 2006.

Mulliner C, Vigna G, Dagon D, Lee W, Using labeling to prevent cross-service attacks against smart phones. In: Proceedings of the detection of intrusions and malware & vulnerability assessment (DIMVA), 2006.

Puttaswamy K, Zhao B, Preserving privacy in location-based mobile social applications. In: Proceedings of the HotMobile, 2010. p. 1–6.

Qi H, Wu D, Khosla P, A mechanism for personal control over mobile location privacy. In: Proceedings of the IEEE/ACM first international workshop on broadband wireless services and applications, BroadWISE, 2004.

Riboni D, Pareschi L, Bettini C, Jajodia S, Preserving anonymity of recurrent location-based queries. In: Proceedings of the 16th international symposium on temporal representation and reasoning, IEEE computer society, 2009.

Schlegel R, Zhang K, Zhou X, Intwala M,Kapadia A, Wang X, Soundcomber: a stealthy and context-aware sound trojan for smartphones. In: Proceedings of the 18th annual network and distributed system security symposium (NDSS), 2011. p. 17–33.

Shin H, Atluri V, Vaidya J, A profile anonymization model for privacy in a personalized location based service environment. In: Proceedings of the 9th international conference on mobile data management. MDM'08, 2008, p. 73–80.

Shokri R, Freudiger J, Hubaux J-P, A unified framework for location privacy. In: Proceedings of the HotPETS, 2010.

Song W, Sean WX, In-device spatial cloaking for mobile user privacy assisted by the cloud. In: Proceedings of the eleventh international conference on mobile data management (MDM), 2010, p. 381–386.

Wang S, Tu G-H, Ganti R, He T, Leung K, Tripp H, et al., Mobile micro-cloud: application classification, mapping, and deployment. In: Proceedings of the annual fall meeting of ITA (AMITA), October, 2013.

Weerasinghe D Rajarajan M, Rakocevic V, Device data protection in mobile healthcare applications. In: Proceedings of the first international conference on electronic healthcare in the 21st century, 2008.

Xu N, Zhang F, Luo Y, Jia W, Xuan D, Teng J, Stealthy video capturer: a new video-based spyware in 3G smartphones. In: Proceedings of the second ACM conference on wireless network security, WiSec '09, New York, NY, USA, 2009, p. 69–78.

⟨http://ec.europa.eu/justice/newsroom/data-protection/news/120125_en.htm⟩, 2012.

⟨http://www.bbc.co.uk/news/business-17192234⟩, 2012.

⟨http://www.bbc.co.uk/news/technology-17178954⟩, 2012.

Chen Yu-Jia, Wang, Li-Chun, A security framework of group location-based mobile applications in cloud computing. In: Proceedings of the 40th international conference on parallel processing workshops (ICPPW), 2011. p. 184–190.

Zhong S, Li L E, Liu Y G, Yang Y R. Privacy-preserving location-based services for mobile users in wireless networks. Technical Report. State University of New York; 2005.