Privacy issues in location-aware browsing

[Position paper]

Maria Luisa Damiani
Department of Informatics and Comunication
University of Milan, Italy
damiani@dico.unimi.it

Pierluigi Perri
School of Law
University of Milan, Italy
pierluigi.perri@unimi.it

ABSTRACT

Advances in positioning services and their pervasiveness, e.g., wi-fi based location services, pave the way to the development of innovative LBSs and architectures. In this paper we focus on location-aware browsing, a framework which enables websites to acquire the position of website users. In particular we discuss privacy issues related to the recent W3C proposal for a geolocation API standard. Such specification prescribes that users must give explicit consent to the disclosure of position information to websites. In this paper we argue that stronger and more flexible protection is needed: a) users should be provided with the capability of disclosing coarse regions in place of point coordinates in order to limit the disclosure of personal location data; b) location information should be protected not only against websites but also against location service providers. We discuss a possible approach to address those requirements under the assumption that the position is computed by a wi-fi based positioning service. Finally, we broaden the discussion to include a complementary legal viewpoint.

Categories and Subject Descriptors

H.2.8 [**Database management**]: Database applications—*Spatial databases and GIS*; K.4.1 [**Computers and society**]: Public Policy Issues—*Privacy*

General Terms

Management, legal aspects, standardization, algorithms

Keywords

Privacy, location-based services, mobility

1. INTRODUCTION

One of the factors that greatly contributes to the rapid development of LBSs over recent years is the availability of novel and cost-effective positioning techniques. Public wi-fi based positioning (WPS) is perhaps the technique which best exemplifies the evolution of positioning technologies beyond GPS. WPS technology is becoming pervasive and market studies (http://www.in-stat.com) project an increase in the number of wi-fi enabled devices from over 500 million in 2009 to nearly 2 billion in 2014. WPSs rely on the existing network infrastructure to allow seamless localization of mobile customers in both indoor and outdoor spaces with an accuracy that is sufficient for most applications. Typically the positioning service is provided by a third party, the *location service provider* (LS), e.g., Google and Apple, which computes the position based on the contextual data sent by the mobile clients, i.e., the access points nearby. The diffusion of WPSs, along with the fact that aggregated location data can be increasingly exploited commercially, contributes to the flourishing of LBSs, architectures and geosocial networks.

Location-aware browsing is a novel framework for location-aware services which enables websites to collect location information about website users. Location data can then be used to provide users with localized information services. These services, unlike conventional on-demand LBSs, can be *pushed* to users, i.e., services are provided even though they are not explicitly requested. Other LBSs based on the push model exist, like for example proximity-based advertising services. What is peculiar in the case under consideration is that the user's position is acquired while that user is browsing the web. Web pages in fact contain scripts which can request the position of the mobile device hosting the implementation. By combining pervasive positioning services with location-aware browsing, websites and LBS providers are potentially able to collect vast amounts of position data. LSs are even in a more favorable situation because they can acquire the position of users across different websites. All this inevitably magnifies the concern for location privacy because the users' location can be easily collected and disclosed to a variety of parties.

Recently the W3C Geolocation Working Group has released a proposal for a standard scripted access interface to location information associated with the hosting device [13]. This proposal is referred to as Geolocation API Specification. Notably the interface is privacy-aware in that the specification prescribes that users must give explicit consent to send the device position to a particular website. However, this solution presents important limits for what concerns privacy protection. A comprehensive analysis of those limits is reported in [6]. In this paper we want to contribute some

SPRINGL'10, November 2, 2010, San Jose, CA, USA
Copyright 2010 ACM 978-1-4503-0435-1/10/11 ...$10.00.

additional considerations to such an analysis. In particular we emphasize the following two requirements:

(a) In the current specification, the location information is disclosed at the finest level of granularity obtainable. Conversely, users may require to send to websites a coarser location in place of an accurate position and in this way minimize the location disclosure and the privacy risk.

(b) The current privacy protection strategy aims to protect the users' location disclosed to websites (and in general LBS providers). However, the positioning services are increasingly provided by third parties, i.e., the LSs, which not necessarily are trusted by users. The question is whether some form of protection can be provided without giving up location-aware browsing.

In summary, privacy protection in location-aware browsing has two faces: (a) protection against websites; and (b) protection against LSs. In this paper we discuss these requirements under the assumption that the positioning service provided by LSs is based on WPS. Then we sketch a possible approach to address those requirements. We first consider the requirements separately and then the case in which both requirements are to be fulfilled.

The paper is organized as follows. Section 2 specifies the problem context. Section 3 outlines possible approaches to tackle the above requirements. Section 4 broadens the discussion to the contextual legal framework. Section 5 ends the paper with some conclusive remarks.

2. THE CONTEXT

We start by providing some background knowledge on WPS. We also report a few considerations on the quality of the service. Then we provide a quick overview of the Geolocation API specification.

2.1 Wi-fi based positioning

2.1.1 Background

WPSs rely on the existing wi-fi network infrastructure, consisting of public and private Access Points (APs). No dedicated infrastructure in needed, the positioning service only requires that the mobile device is wi-fi enabled and connected to Internet. We borrow the terminology from Skyhook Wireless[1] to illustrate the general WPS architecture in Figure 1. The system consists of a set of wi-fi enabled mobile devices (called Location Nodes, LN), a set of APs and a Lookup Table (LLT). The LLT is the database reporting the association between the APs and their physical position. The LLT is commonly handled by the LS. The process for the determination of the position consists of two main steps. In the first step, the LN gathers relevant data about the APs in the vicinity. APs broadcast beacon frames carrying the Medium Access Control (MAC) address (i.e., the AP identifier) and additional information such as the name of the network (SSID). In the subsequent step, the LN transmits the set of observed APs to the LLT. Their MAC address is thus matched against the database. If the matching is successful, then the coordinates of the APs are used to determine the position of the LN which is then returned to the requester. Conversely, if the matching is not successful a coarser estimation of the position is usually made based for example on IP address.

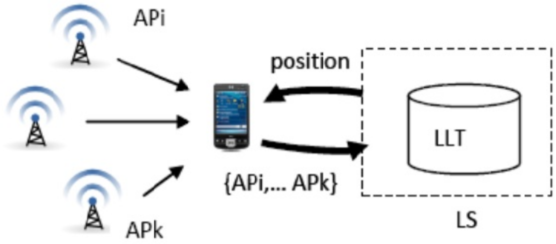

Figure 1: Positioning process

For the sake of concreteness, we report below a fragment of the protocol adopted by Google Gears, a platform which provides geolocation capabilities to a few browsers, such as Firefox 3.5+, through a high level scripting language [2]. In this example, the contextual information which is transferred to the LS (i.e., Google Location Service) includes the field "wifi_towers" specifying the list of observed APs, each identified by the six-bytes MAC Address, along with the signal strength and an additional attribute.

```
{ "version": "1.1.0",
  "host": "maps.google.com",
..........................
  "wifi_towers": [  {
      "mac_address": "01-23-45-67-89-ab",
      "signal_strength": 8,
      "age": 0},
    { "mac_address": "01-23-45-67-89-ac",
      "signal_strength": 4,
      "age": 0} ]}
```

2.1.2 The content of the LLT table

The content of the LLT affects the accuracy and reliability of the position information. For example, if existing APs are not registered in the database, it may happen that the position cannot be returned at the best accuracy, while if the coordinates of an AP are not correct then the returned position may be unreliable. The properties of LLT completeness and soundness are important to evaluate the quality of the positioning service. Yet, very limited information is commonly available.

The LLT typically contains millions of records. Various strategies are currently adopted to populate the LLTs. For example Skyhook Wireless populates the database through a process known as *wardriving*, namely a professional team of drivers drive around using properly equipped cars and recording observed APs and cell towers along with the corresponding GPS readings. Google uses a similar technique with the fleet of Streetview vehicles. Navizon applies a collaborative approach to mapping (*crowdsourcing*), that is, the position of APs and cell towers are provided by registered end users who are somehow rewarded for the collected data[3]. Another database which is fed by users on a volun-

[1] http://www.skyhookwireless.com/

[2] http://code.google.com/apis/gears/geolocation/network_protocol.html

[3] http://www.navizon.com/

tary basis is the open database Wigle [4]. In all these cases, we are not aware of any indicator of the size and quality of the database apart from few statistics reported by Wigle.

The map in Figure 2 illustrates the result of an experiment that we conducted in early 2010 on a small region close to the Milan metropolitan area, using non professional wardrivers to analyze the quality of a LLT on that area [5]. The map reports the position of both the APs registered in the LLT (yellow pinpoints) and those that have been observed during in-house wardriving (orange pinpoints). All the pinpoints are reported on a Google Earth map and each of them is labeled with the MAC Address of the corresponding AP and the SSID of the related network. We have observed that in most cases, whenever an AP is reported in the LLT, the distance from the actual position is within the tolerance threshold (i.e., within 150 meters). If so, the coordinates in the database can be considered correct. In certain cases, however, the actual position and the corresponding registered position are far from each other. An example of inconsistent positions is reported in Figure 2: the elements underlined in green refer to the same AP but there is significant distance between them. Such a discrepancy means that the coordinates reported in the database are very likely not reliable.

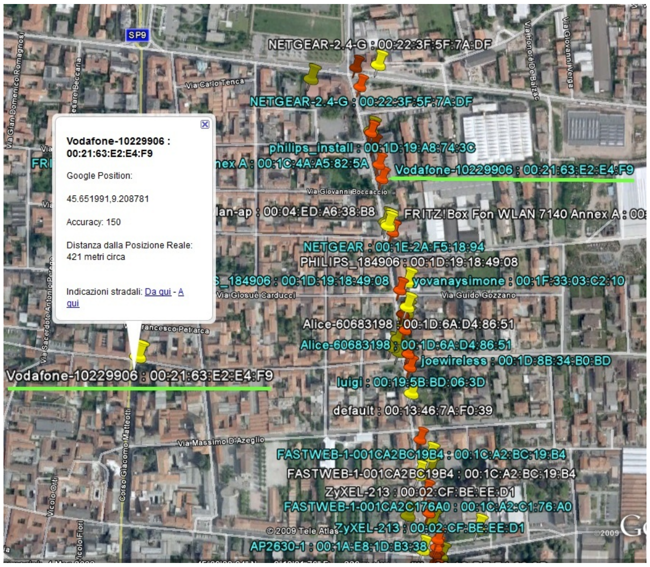
SP9
NETGEAR-2.4-G : 00:22:3F:5F:7A:DF
Via Carlo Tenca
NETGEAR-2.4-G : 00:22:3F:5F:7A:DF
Vodafone-10229906 :
00:21:63:E2:E4:F9
Google Position:
45.651991,9.208781
Accuracy: 150
Distanza dalla Posizione Reale:
421 metri circa
Indicazioni stradali: Da qui - A qui
philips_install : 00:1D:19:A8:74:3C
nex A : 00:1C:4A:A5:82:5A
Vodafone-10229906 : 00:21:63:E2:E4:F9
Via Giovanni Boccaccio
lan-ap : 00:04:ED:A6:38:B8
FRITZ!Box Fon WLAN 7140 Annex A : 00:
NETGEAR : 00:1E:2A:F5:18:94
PHILIPS_184906 : 00:1D:19:18:49:08
S_184906 : 00:1D:19:18:49:08
yovanaysimone : 00:1F:33:03:C2:10
Via Giosuè Carducci
Via Guido Gozzano
Alice-60683198 : 00:1D:6A:D4:86:51
Alice-60683198 : 00:1D:6A:D4:86:51
joewireless : 00:1D:8B:34:B0:BD
Vodafone-10229906 : 00:21:63:E2:E4:F9
luigi : 00:19:5B:BD:06:3D
default : 00:13:46:7A:F0:39
Via Massimo D'Azeglio
FASTWEB-1-001CA2BC19B4 : 00:1C:A2:BC:19:B4
FASTWEB-1-001CA2BC19B4 : 00:1C:A2:BC:19:B4
ZyXEL-213 : 00:02:CF:BE:EE:D1
FASTWEB-1-001CA2C176A0 : 00:1C:A2:C1:76:A0
ZyXEL-213 : 00:02:CF:BE:EE:D1
AP2630-1 : 00:1A:E8:1D:B3:38
© 2009 Tele Atlas
©2009

Figure 2: The APs observed on field (orange) vs. the APs reported in the database (yellow)

2.2 Geolocation API specification

The Geolocation API specification ignores the details of the positioning technologies. In this sense the specification is agnostic of the underlying location information sources, which include, for example, GPS and GSM besides WPS. The specification consists of two main parts: (a) a programming interface; (b) normative (i.e., binding) requirements for the user agents (web browser) and the requesting websites. Additional non-normative requirements are not relevant to this discussion. The programming interface consists of a set of objects that run on the host device to compute the device location. In particular the interface provides three high level functions: *GetPosition* returns the current position of the device or an error; *WatchPosition* and *ClearWatch* are respectively to repeatedly report the updated position of the device and to stop such updating. Note that these functions do not require as input the name or a reference to LS. Regarding the privacy aspect, the fundamental normative requirement prescribes that the user agent must, in most cases, get the user's consent to send the device's location to a particular website before initiating a process to obtain a cached or new location [6]. Upon the request of consensus, the user can respond yes or no. In case of positive response, the position at the finest granularity obtainable is sent to the website. The user however is not explicitly informed of the LS computing the user's position.

3. THE TWO FACES OF PRIVACY

3.1 Protecting location against websites

As we have pointed out, in the current version of the specification, the position is returned to the website at the finest granularity obtainable. Therefore end users cannot choose to send websites a coarser region in place of more precise positions. This limitation has been emphasized in [6] which finally recommends a more flexible solution, in which the user can choose the granularity from a predefined hierarchy of granularities, e.g., street, zip code, city, or by creating custom cloaked regions.

Indeed the definition of a standard location hierarchy is the simplest solution and is currently adopted by several commercial LBSs platforms, e.g., Yahoo FireEagle. This approach, however, presents a number of shortcomings: first, cloaked regions can be too small to effectively blur the user's position or conversely too broad, say a zip code area of several square kilometers, to preserve the utility of the location information. In addition this solution does not protect against location inferences. For example, it can be shown that under certain circumstances, e.g., the speed is known, the location of tracked users inside coarse regions can be inferred from the observation of the movement across the regions of space [8]. Moreover, if all the positions are indistinctly blurred, the usefulness of certain applications like location-sharing applications is compromised, because for example users cannot let their acquaintances know the exact position, unless explicitly changing the privacy options [4].

Compared with this solution, custom coarse regions are definitely more flexible and potentially assure a more effective protection against location inferences. Coarse regions are generated by *location cloaking algorithms.* A cloaked region may have a regular shape (i.e., rectangle, circle) or an irregular one. A broad literature exist on cloaking algorithms [10, 2]. At the current stage, however none of the those methods can be considered the "best". Moreover, location privacy has different facets, for example privacy can regard the geometric position (i.e., coordinates (x,y)) or the semantic position (i.e., places like hospitals and religious buildings [3, 5]), and for each of these aspects a number of solutions exist. Therefore, any cloaking solution relying on a specific method would be arbitrary. A preferable approach to is to allow a flexible interface in which the cloaking service is one of the input parameters of the operation computing the po-

[4] http://www.wigle.net

[5] We refer to the LLT used by Google Location Service

sition. In that way the website can request the positioning service to process the position before returning it. We believe that such a flexibility and openness towards cloaking methods is definitely needed to allow competing LSs to provide more effective privacy protection solutions.

3.2 Protecting location against LS

To comply with existing privacy regulations, LSs specify in their privacy policies which data are collected and how those data are used. For example, in a privacy policy we can read [6] that the information collected by the LS consists of the wi-fi routers, cell ids of the cell towers, the strength of wi-fi or cell signal, IP address, user agent information, and unique identifier of the client. Moreover: " The information collected above will be anonymized and aggregated before being used.....This means that your IP address and unique identifier of your client will be stripped out before being used...".

It is well known, however, from research literature that the simple removal of users' identifiers is not enough to protect identities, because other attributes can exist, called *quasi-identifiers*, that linked with additional information can lead to the identification of the individual [12]. Similarly, it is well-known that personal location is a quasi-identifier [9]. For example if the position of an individual falls into a residential building at night, it is likely that such a building is the home of the individual, and thus the user's identity can be easily discovered.

Since the pioneering work of Gruteser et al. [9] and Beresford et al. [1], a large number of solutions have been developed to prevent undesirable inferences that can defeat the privacy protection mechanism and reveal the hidden identity or location (see [10, 2, 7, 11] for a survey). To our knowledge, however, none of those techniques are deployed in real mobile applications. As a consequence, the location data gathered by LSs are very likely exposed to privacy leaks or at least users can perceive this risk.

The question we pose is whether users can benefit from location-aware browsing and in general of LBSs without necessarily disclosing the position to the LS. Of course if positions were determined by the mobile device, for example based on GPS signal, the problem would not exist because no third party would be required. We recall, however, that positioning is based on WPS and that WPSs rely on the use of a database (i.e., the LLT table) which is normally proprietary and has a very large size, and thus cannot be replicated locally. Therefore the position cannot be obtained unless interacting with the LS. Put in these terms, the problem does not seem to have solutions.

We thus propose a different formulation according to which the problem becomes "to compute the position at the quality provided by a LS without letting the LS know the user's position *every time* the position is needed". In other terms, the idea is to minimize the interaction with the LS so as to limit the amount of personal data flowing towards the LS. Below we provide some hints for a possible approach to the problem.

3.2.1 Outline of a possible approach

We observe that individuals generally spend significant amount of time within indoor environments. Moreover, individuals tend to frequent the same places, e.g., home, office, shops, recreational places. In the light of these considerations, a possible approach to the minimization of location requests is to enable some form of position caching. The idea is to store location information in a cache on the mobile device and thus interact with the LS only when the requested information is not present in the cache. A key question is what kind of information is to be recorded in the cache.

We recall that in our setting the position is returned by a WPS and that in such a case the position is computed based on the contextual data sent by the mobile device, i.e., the *pattern* of APs where a pattern identifies a set of APs along with possibly additional parameters such as the strength of the AP signal. A possible approach is to store in the cache subsets of the LLT. The drawback is that portions of the proprietary database are to be replicated locally. A more appealing solution is to cache on the mobile device the association between APs patterns and positions so as to maintain the historical data against which to match subsequent requests. Upon a location request, the APs in the vicinity are detected and thus the APs pattern is searched in the cache. If the matching is not successful, the location is requested to the LS. In this way the interaction with the LS takes place only when the local information is not sufficient, as we required. An intriguing question is how to define a suitable matching operation. The operation is complex because, in areas at high APs density, the same position can be returned by many different patterns.

3.3 Combining the strategies

An additional step towards a stronger privacy protection is to provide coarse locations to websites while limiting the interaction with the LS. A possible direction of research is to combine the two previous strategies. In essence the idea is to cache coarse regions, in place of coordinates. Consider a cache recording triples $< p, cr, g >$ where p is the APs pattern, cr the cloaked region and g the cloaking service, e.g., the location granularity. A possible direction of research is to consider the following protocol: initially the mobile device issues the location request by specifying the desired cloaking service g and the APs pattern p. Thus, if the pair (p, g) is not in the cache, the request is passed to the positioning service which returns a coarse region cr. The cache is then updated with the triple <p, cr, g>. Therefore, when a new request is made from a position which is sufficiently close to the previous one, the device can likely find the cloaked region directly in the cache and then return cr. In this way the website or even the user can specify the desired "level of location privacy" while the communication with the LS can be limited.

4. LEGAL PERSPECTIVE

Privacy requirements arise from a variety of sources, including legislation. Therefore it seems useful to highlight, in addition to the technical issues, the legal dimension of privacy. In this section we overview the legal framework within the European legislation related to the regulation applicable to LBSs. First of all, the European law through the Directive 95/46/EC distinguishes between *personal data* and *sensitive data*. Personal data is "any information relating to an identified or identifiable natural person...". Sensitive data are personal data which can reveal "racial or ethnic origin, political opinions, religious or philosophical beliefs, trade-

[6] http://www.google.com/privacy-lsf.html

union membership, and the processing of data concerning health or sex life". The treatment of sensitive data is usually subject to more severe restrictions by law than ordinary personal data.

A first question is where location data fit into this classification. The Article 29 Working Party, an organization grouping the Data Protection Authorities of the UE countries, states that "Since location data always relate to an identified or identifiable natural person, they are subject to the provisions of the protection of personal data laid down in Directive 95/46/EC, even if not all the location data are related to a subject but can identify also and object not directly linkable with a person." Accordingly, location information is to be considered personal data and must be subject to the security and privacy requirements provided by European and national laws on data protection. An open question is whether location data can be also sensitive because that is not clearly stated.

Another classification related to geo-location privacy is reported in the more recent Directive 2002/58/EC, a special Directive for e-communications[7], which contains the definition of *location data* and *traffic data* where the latter is "any data processed for the purpose of the conveyance of a communication on an electronic communications network or for the billing thereof". Traffic data often contain geolocation information, for example where a call started and ended. Accordingly, LBSs should be considered systems for personal data processing. Conversely, based on the aforementioned Directive, these data could be treated either as location data or traffic data or both.

To conclude this brief overview, it is important to outline that the Directive 2002/58/EC contains additional provisions related to the processing of traffic data[8] and in particular to: confidentiality of the communications; traffic data processing rules; and regulation of location data other than traffic data. The legal principles relevant for LBSs are drawn from those articles:
- Confidentiality. Member States shall prohibit listening, tapping, storage or other kinds of interception or surveillance of communications and the related traffic data by persons other than users, without the consent of the users concerned, except when legally authorized to do so in accordance with Article 15 of the Directive.
- Necessity. Traffic data relating to subscribers and users processed and stored by the provider of a public communications network or publicly available electronic communications service must be erased or made anonymous when it is no longer needed for the purpose of the transmission of a communication except in some specific cases.
- User consent/ data anonymization. Where location data other than traffic data, relating to users or subscribers of public communications networks or publicly available electronic communications services, can be processed, such data may only be processed when they are made anonymous, or with the consent of the users or subscribers to the extent and for the duration necessary for the provision of a value added service.

5. CONCLUSION

In this paper we have discussed privacy issues in emerging location-aware browsing solutions and we have prospected possible directions of research. The protection of location privacy has however a dual dimension, technological and legal. That paves the way to challenging research along both directions.

6. REFERENCES

[1] A. R. Beresford and F.Stajano. Location Privacy in Pervasive Computing. *IEEE Pervasive Computing*, 2(1):46–55, 2003.

[2] C. Bettini, S. Mascetti, X. S. Wang, D. Freni, and S. Jajodia. Anonymity and historical-anonymity in location-based services. In *Privacy in Location-Based Applications*, pages 1–30, 2009.

[3] M.L. Damiani, E. Bertino, and C. Silvestri. The PROBE Framework for the Personalized Cloaking of Private Locations. *Transactions on Data Privacy*, (3)2:123–148, 2010.

[4] M.L. Damiani, C. Silvestri, and E. Bertino. Fine-grained cloaking of sensitive positions in location sharing applications. IEEE Pervasive Computing (accepted for publication).

[5] M.L. Damiani, C. Silvestri, and E. Bertino. Analyzing semantic location cloaking techniques in a probabilistic grid-based map. In *Proc. ACM GIS 2010 (demo)*, 2010.

[6] N. Doty, D. Mulligan, and E. Wilde. Issues of the W3C Geolocation API. Technical report, UC Berkeley, School of Information, 2010.

[7] M Duckham and L. Kulik. Location privacy and location aware computing. In *Drummond J (ed) Dynamic & mobile GIS: investigating change in space and time. Boca Raton. CRC Press*, 2006.

[8] G. Ghinita, M.L. Damiani, C. Silvestri, and E. Bertino. Preventing Velocity-based Linkage Attacks in Location-Aware Applications. In *Proc. of the 17th ACM GIS*, 2009.

[9] M. Gruteser and D. Grunwald. Anonymous Usage of Location-Based Services Through Spatial and Temporal Cloaking. In *Proc. of the 1st International Conference on Mobile systems, Applications and Services.* ACM Press, 2003.

[10] C. S. Jensen, H. Lu, and M.L. Yiu. Location Privacy Techniques in Client-Server Architectures. In *Privacy in Location-Based Applications: Research Issues and Emerging Trends*. Springer-Verlag, 2009.

[11] J. Krumm. A survey of computational location privacy. *Personal and Ubiquitous Computing*, (13)6:391–399, 2009.

[12] L. Sweeney. Achieving k-anonymity privacy protection using generalization and suppression. *Int. Journal on Uncertain. Fuzziness Knowl.-Based Syst.*, 10:571–588, 2002.

[13] W3C. Geolocation api specification. http://dev.w3.org/geo/api/spec-source.html, 2010.

[7] It overrides the general Directive 95/46/EC whenever they overlap

[8] Articles 5, 6 and 9 of Directive 2002/58/EC