How Others Compromise Your Location Privacy: The Case of Shared Public IPs at Hotspots

Nevena Vratonjic[1], Kévin Huguenin[1],
Vincent Bindschaedler[2,⋆], and Jean-Pierre Hubaux[1]

[1] School of Computer and Communication Sciences, EPFL, Switzerland
[2] Department of Computer Science, UIUC, USA

Abstract. Location privacy has been extensively studied over the last few years, especially in the context of location-based services where users purposely disclose their location to benefit from convenient context-aware services. To date, however, little attention has been devoted to the case of users' location being unintentionally compromised *by others.*

In this paper, we study a concrete and widespread example of such situations, specifically the location-privacy threat created by access points (e.g., public hotspots) using network address translation (NAT). Indeed, because users connected to the same hotspot share a unique public IP, a single user making a location-based request is enough to enable a service provider to map the IP of the hotspot to its geographic coordinates, thus compromising the location privacy of all the other connected users. When successful, the service provider can locate users within a few hundreds of meters, thus improving over existing IP-location databases. Even in the case where IPs change periodically (e.g., by using DHCP), the service provider is still able to update a previous (IP, Location) mapping by inferring IP changes from authenticated communications (e.g., cookies).

The contribution of this paper is three-fold: (i) We identify a novel threat to users' location privacy caused by the use of shared public IPs. (ii) We formalize and analyze theoretically the threat. The resulting framework can be applied to any access-point to quantify the privacy threat. (iii) We experimentally assess the state in practice by using real traces of users accessing Google services, collected from deployed hotspots. Also, we discuss how existing countermeasures can thwart the threat.

1 Introduction

With the ubiquity of mobile devices with advanced capabilities, it is becoming the norm for users to be constantly connected to the Internet; users can benefit from many online services while on-the-go. Among others, location-based services (LBSs) are increasingly gaining popularity. With an LBS, users share their location information with a service provider in return for context-aware services, such as finding nearby restaurants. Users also enjoy sharing location information with their friends on social networks (e.g., Facebook and Twitter) [26].

[⋆] Parts of this work were carried out while Vincent Bindschaedler was with EPFL.

E. De Cristofaro and M. Wright (Eds.): PETS 2013, LNCS 7981, pp. 123–142, 2013.

Although very convenient, the usage of LBSs raises serious privacy issues. Location privacy is a particularly acute problem as location information is valuable to many parties, because much of information can be inferred from users' locations (e.g., users' interests and activities). Location information is essential for many online service providers [13], especially for those whose business models revolve around personalized services. A prominent example is online advertising, an ever-increasing business with large revenues (e.g., $22.4 billion in the US in 2011 [28]), as so-called location-specific ads based on the location information are significantly more appealing to users [20].

Typically, users willingly disclose their location only to LBS providers. Yet, non-LBS providers can obtain users' locations through *IP-location*: determining the location of a device from its IP. Existing IP-location services rely either on (i) active techniques, typically based on network measurements [21], or (ii) passive techniques, consisting of databases with records of IP-location mappings [19, 24]. Active techniques provide more accurate results, however they incur high measurement overhead and a high response time (in the range of several seconds to several minutes) to localize a single IP. A passive approach is usually much faster and thus preferred by service operators. A number of IP-location databases are available, either free (e.g., HostIP [19]) or commercial (e.g., MaxMind [24]). Some databases contain records of landmark IPs for which the location can be inferred (e.g., institutions [37] or websites that post their location [17]) and other IPs are geolocated relatively to these landmarks. However, they provide at most a city-level accuracy and most of the entries refer only to a few countries [27]. For instance, MaxMind reports to correctly geo-locate, within a radius of 40 km, 81% of IP addresses in the US and 60%-80% in Europe. This level of accuracy is effective for regional advertising but is not sufficient for local businesses (e.g., bars) which require neighborhood or street-level accuracy [20]. Thus, major Web companies, including Google, are actively working on improving IP-location[1].

Service providers can also obtain a user's location via transitivity, relying on users to disclose their location and that of others in their vicinity: if a provider knows the location of user B and that user A is close to B, the provider knows roughly the location of A. Such situations arise when users report neighboring users (e.g., Bluetooth), or *check-in* on online social networks (OSNs) and tag friends they are with. In some cases, even if the proximity information is not directly revealed by users, a provider can still infer it, as we will show.

In this paper, we study a location-privacy threat users are exposed to on a daily basis. When a user connects to the Internet through the same access point (AP) as other users (e.g., a public hotspot, home router) who make LBS queries, the service provider learns the user's location. Indeed, because all of the devices connected to a public hotspot, implementing network address translation, share the AP's public IP, when users generate LBS queries, the service provider learns the location of the AP and maps it to the AP's public IP. IPs remain the same

[1] Google reports an accuracy of 95% at the region-level and 75% at the city-level, with high variance across countries, and seeks to improve it to the street-level [14].

for a certain amount of time, thus for any connection for which the source IP is the same as the AP's IP, the service provider can conclude that the device is located nearby the location of the AP. The accuracy of the estimated location depends on the range of the AP (typically under one hundred meters) and on the accuracy of the locations reported by users in LBS queries (typically under ten meters with GPS-geolocation). Thus, it is significantly more accurate than the existing IP-location databases. The fact that the threat is based on observing the user's IP, which might be inferred, e.g., by using a Java applet [25], even when the client tries to hide it, makes the threat even more difficult to evade.

The (IP, Location) mapping the adversary obtains for the AP stays valid until the IP changes. Dynamic IP addresses (provided that IPs are allocated to geo-diverse hosts), short DHCP leases, and systematic assignment of new IPs upon DHCP lease expiration therefore have a positive effect on location privacy. However, even when the IP is renewed and changes, service providers have means to learn about the IP change, for example, due to the widespread use of *authenticated* services (e.g., e-mails, OSNs). Consider a user connected to the AP who checks her e-mail shortly before and after an IP change. As a unique authentication cookie is appended to both requests, the service provider can conclude that the same user has connected with a new IP and can therefore update the (IP, Location) mapping with the new IP. In fact, it is sufficient that the service provider is able to link the requests to the same user, based on cookies, user agent strings, or any fingerprinting technique, e.g., [39].

The contribution of this paper is three-fold: (i) We identify the location-privacy threat that arises from the use of shared public IPs. Because the problem is inherent in the way networks (i.e., NAT) operate and its wide deployment, the potential impact of the threat is significant. The expected accuracy of locating affected users is about few hundreds of meters. (ii) We formalize and analyze the problem theoretically and we provide a framework to estimate the location-privacy threat, namely the probability of a user being localized by a service provider. The framework is easily applicable to any access point setting: it employs our closed-form solution and takes as input an AP's parameters (i.e., a few aggregated parameters, such as user connection and traffic rates, that can be extracted from logs) and it quantifies the potential threat. It is a lightweight alternative to extensive traffic analysis. The framework thus constitutes a valuable input to model sporadic location exposure. (iii) We evaluate experimentally the scale of the threat based on real traces of users accessing Google services, collected for a period of one month from deployed hotspots. Even at a moderately visited hotspot, we observe the large scale of the threat: the service provider, namely Google, learns the location of the AP only about an hour after users start connecting and within 24 hours he can locate up to 73% of the users. Finally, we discuss how existing countermeasures could thwart the threat. To the best of our knowledge, this is the first paper that addresses the problem of users' locations being exposed by others at NAT access points.

2 Background

In this section, we provide relevant background on the technical aspects underlying the considered problem.

IPv4 (public) Address Allocation. To communicate on the Internet, hosts need public IP addresses. An IP can be either *static*, i.e., permanently fixed, or *dynamic*, i.e., periodically obtained from a pool of available addresses, typically through the Dynamic Host Configuration Protocol (DHCP). Dynamic IP is used for a limited amount of time specified by the *DHCP lease*. For convenience, upon DHCP lease expiration, hosts are often re-assigned the same IP. A large-scale study shows that over one month, less than 1% of the hosts used more than one IP and less than 0.07% used more than three IPs [4]. More than 62% of dynamic IPs on average remain the same over a period of at least 24 hours [38].

Network Address Translation (NAT). NAT hides an entire IP address space, usually consisting of private IPs, behind one or several public IPs. It is typically used in Local Area Networks (LANs), where each device has a private IP, including the gateway router that runs NAT. The router is connected to the Internet with a public IP assigned by an ISP. As traffic is routed from the LAN to the Internet, the private source IP in each packet is translated on-the-fly to the public IP of the router: traffic from all of the hosts in the LAN appears with the same public IP–the public IP of the NAT router. A study shows that about 60% of users are behind NATs [4].

Geolocation. Mobile devices determine their positions by using their embedded GPS or an online geolocation service. With a GPS, the computation takes place locally by using satellites' positions and a time reference. Commercial GPS provides highly accurate results (< 10 meters) [35], especially in "open sky" environments. With online geolocation services (e.g., Skyhook) a device typically shares the list of nearby cell towers and Wi-Fi APs together with their signal strengths, based on which the server estimates the device's location by using a reference database. Such databases are built mostly by GPS-equipped mobile units that scan for cell towers and Wi-Fi APs and plot their precise geographic locations. Inputs of users with GPS-equipped devices, who provide both their positions and the surrounding stations, are also taken into account. Reported accuracy of such systems is about 10 meters [32].

3 System Model

In this section, we elaborate on the considered setting, notably NAT access points, the location-privacy threat, and the adversary.

3.1 Setting

We consider a *NAT Access Point* setting, a prevalent network configuration, where users connect to the Internet through an access point (AP), such as a public hotspot, a home (wireless) router or an open-community Wi-Fi AP (e.g., FON), as depicted in Fig. 1. An AP, located at (x_1,y_1), is connected to the Internet by a given ISP and provides connectivity to the authorized users. The AP has a single *dynamic public* IP that is allocated with DHCP by the ISP: The AP's public IP is selected from a DHCP pool of available IPs and is valid during the DHCP lease. When connecting to the AP, each device is allocated a *private* IP and the AP performs network address translation (NAT). Consequently, on the Internet, all connections originating from the devices connecting through the AP have the same source IP, which is the public IP of the AP.

While connected to the Internet through an AP, users make use of various online services including search engines, e-mail, social networks, location-based and online geolocation services. Services can be used either in an authenticated (e.g., e-mail) or unauthenticated way (e.g., search). We consider that the requests a server receives from the devices connected to the AP are of the following types:

1. Geolocation requests: `Geo-Req(`MACs`)`, where MACs refer to the MAC addresses of the APs and cell towers in the range of the device;
2. LBS requests: `LBS-Req(`(x_0, y_0)`)`, where (x_0, y_0) denotes the coordinates of the device[2] (assumed close to the AP's location (x_1, y_1)) shared by the user;
3. Authenticated standard (i.e., that are neither LBS nor Geolocation) requests: `Auth-Req(`*tok*`)`, where *tok* represents any information that allows for user authentication or linkability of user requests (e.g., a cookie or a username);
4. Unauthenticated standard requests: `Req()`.

With LBS requests, the service provider obtains the user's location under several forms and by different means. The user can specify her location in free-text (e.g., "bars close to Park and 57th, NYC") or by pin-pointing her location on a map. The location can also be determined by the user's device using one of the techniques described in Section 2 and communicated to the service provider by a mobile application or by her browser through the HTML 5 `getCurrentPosition` JavaScript function. Note that non-LBS applications and websites might access the user's location as well.

Both `Geo-Req` and `LBS-Req` contain an estimate of the AP's coordinates, thus they both enable the server to build the (IP, (x_1, y_1)) mapping. Consequently, there is no need to distinguish between these two types of requests, and we simply refer to both as LBS requests. For all types of request, the server knows the source IP, specifically the AP's public IP.

3.2 Adversary and Threat Models

We consider an adversary whose goal is to learn users' current locations, for instance, to make a profit by providing geo-targeted (mobile) ads and recommendations (e.g., a private company). The adversary has access to the information

[2] We assume that all LBS requests concern users' actual locations, or that the server has means to distinguish between such requests and other LBS requests. It is the case when the location is obtained directly (see Section 2), and sent to the server.

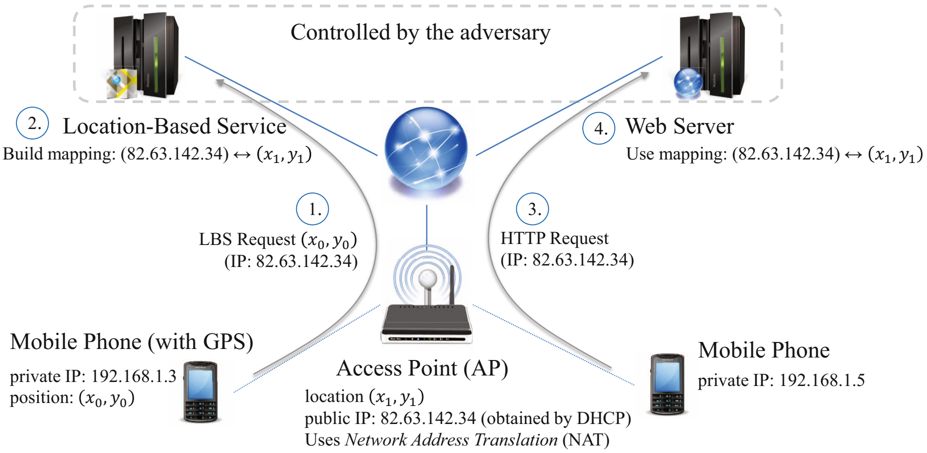


Fig. 1. System and threat model. Devices connect through a NAT access point and share a public IP. A user making an LBS request reveals her location (close to the AP) to the adversary (1) who then builds the (IP,Location) mapping (2). When another user connects to a different server controlled by the adversary (3), the adversary uses the (IP,Location) mapping to locate her as she connects with the same IP (4).

collected by a number of servers that provide online services described above. Companies, such as Google for instance, provide Web search (Google), e-mail (GMail), social networking (Google+), and geolocation and location-based services (Maps). As such, it receives requests of the four types and consolidates the information obtained [15]. The extent to which these services are used is exacerbated by their deep integration in the Android operating system. In addition, Google has an advertising network and thus has a strong incentive to obtain and monetize information about users' locations. As a matter of fact, Google is actively working on improving its IP-location based on users' traffic, in particular by mining queries associated with location (e.g., "best burgers NYC") [14].

Microsoft (with Bing, Hotmail, Bing Maps, and Windows Phones) and Apple (with iCloud and iPhone) are other relevant potential candidates for the considered adversary. Besides these major companies, an alliance of service providers can be envisioned to jointly build an IP-location database: each provider contributes IP-location records of its visitors with known locations and benefits from the database for the IPs of users connecting from unknown locations. This joint effort can be coordinated by an ad network that is common to the participating service providers. This approach extends the potential of the threat as it increases the set of potential adversaries: it alleviates the need for each service to receive all three types of requests and a significant fraction of user traffic.

In this paper, we focus on the case where the adversary has access to all four types of requests. The adversary is assumed to be *honest-but-curious*, meaning that he passively collects information but does not deviate from the specified protocol (e.g., implementing active techniques to retrieve users' locations).

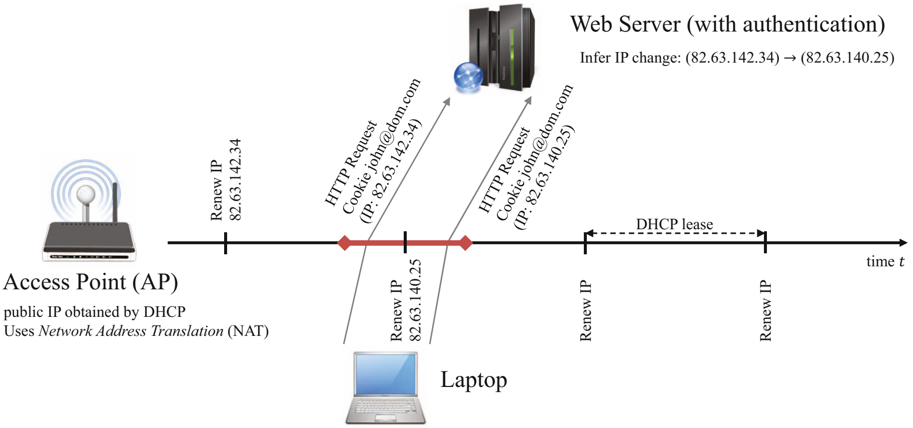


Fig. 2. AP's IP address renewal and updating of the (IP, Location) mapping. A user generates an authenticated request (with a unique cookie) during a DHCP lease interval in which the adversary has obtained the (IP, Location) mapping, shortly before the DHCP lease expires and the AP is assigned a new IP. Shortly after the IP change, the same user generates another authenticated request (with the same cookie) from the new IP. As both requests occurred in a short time interval, the adversary can infer that the AP's IP changed from `82.63.142.34` to `82.63.140.25` and update the mapping.

Given such an adversarial model, we consider the threat of the adversary who learns the location of a user without it being explicitly disclosed: The threat comes from the fact that the adversary can build mappings between an AP's IP and its geographic coordinates based on LBS requests he receives from other users connected to the AP. Because all requests (from devices connected through the AP) share the same public IP, the adversary can subsequently infer the location of the other users. More specifically, considering the example depicted in Fig. 1, when the LBS provider's server (controlled by the adversary) receives an LBS request for position (x_0, y_0), which is the actual position of the user (located close to the AP) determined by her GPS-equipped mobile phone, the server can map the AP's public IP (i.e., `82.63.142.34`) to the approximated AP's location (i.e., $(x_1, y_1) \approx (x_0, y_0)$). Note that the accuracy of the AP's estimated location depends on the GPS accuracy of the user-reported location and the range of the AP. Later, when another user, connected through the AP, makes a request to a server (also controlled by the adversary), then the adversary exploits the obtained mapping and infers from the source IP that the second user is at the same location (i.e., (x_1, y_1)). The adversary can subsequently provide geo-targeted ads. If the adversary is interested in tracking users, he can locate any user who makes an authenticated request before the IP changes.

We assume that the IP addresses in the DHCP pool can be assigned to clients at very distant locations [10]. For instance, some nation-wide ISPs (e.g., SFR in France) assign IPs among the whole set of their clients scattered all over the country. Consequently, the fact that the AP's public IP is dynamic limits in time the extent of the threat: If the AP is assigned a new IP by the ISP, the mapping built by the adversary becomes invalid, unless he is able to infer the IP change. The inference can be based on authenticated requests as depicted in Fig. 2: A request, authenticated by cookie `john@dom.com` and originating from IP `82.63.142.34`, is shortly followed by a request authenticated by the same cookie but originating from a different source IP (i.e., `82.63.140.25`). There are two options: either the AP's IP has changed or the user has moved and is now connected from a different AP. If the inference time interval (delimited with diamonds in Fig. 2) around the IP renewal is short enough, then the adversary can infer, with high confidence, that the IP has changed and its new value.

In summary, the problem we study is as follows. Considering a single AP, time is divided into intervals corresponding to DHCP leases, during which the AP's public IP address remains the same. At a certain point in time, the adversary knows the location of the AP associated to the IP because (i) a user made an LBS request earlier in the time interval or (ii) the adversary knew the location corresponding to the public IP address from the previous interval **and** a user made an authenticated request shortly before and after the public IP address was renewed. The location-privacy threat is to be evaluated with respect to the number of users whose locations are known by the adversary. In the case of geo-targeted mobile ads, the adversary needs to know the location of the user *when* the user makes a requests: the victims are therefore the users who make a standard request *after* the adversary learns the (IP, Location) mapping (during the same DHCP lease). If the adversary is interested in tracking users, he can maintain a log of the users who connected during a DHCP lease and sent requests, and locate them *a posteriori* if he learns the (IP, Location) mapping at some point during the same DHCP lease: the victims are the users who make an authenticated request *during* a DHCP lease in which the adversary learns the (IP, Location) mapping. In this paper, we evaluate the threat with respect to an adversary who aims to exploit *current* location information through geo-targeted ads. However, it is possible to mount more powerful attacks on users' privacy (e.g., track users over time) based on the identified threat.

4 Formalization and Analysis

In this section, we model the aforementioned setting and we build a framework to quantify theoretically the location-privacy threat.

4.1 Model

We consider an access point AP, an honest-but-curious adversary $\mathcal{A}$, and a set of users who connect to AP and make requests to servers controlled by $\mathcal{A}$. We study the system over the continuous time interval $[0, +\infty)$. At each time instant t, AP has a single public IP. Every T time units, starting at time 0, the DHCP lease expires and AP is either re-assigned the same IP or allocated a new one. We model this by independent random variables drawn from a Bernoulli distribution:

with probability p_{New} AP is assigned a new IP, and with probability $1 - p_{\text{New}}$ it is re-assigned the same IP. We divide time into successive sub-intervals I_k, $k \geq 0$, of duration T, corresponding to the DHCP leases: $I_k = [kT, (k+1)T]$. Without loss of generality, we assume the duration of IP leases to be constant. Each sub-interval is aligned with a DHCP lease. Therefore, within each sub-interval AP's public IP remains unchanged. For any time instant t, we denote by $\bar{t}$, the relative time within the corresponding sub-interval, that is $\bar{t} = t \mod T$.

Users connect to AP, remain connected for a certain time and then disconnect. While connected, users make requests, each of which is of one of the following types: LBS, authenticated, or standard. All modeling choices in this section follow well-established conventions [30]–e.g., Poisson processes are known to fit well users arrival and access to services–and are backed up by several public Wi-Fi hotspot workload analysis (e.g., [11]). In addition, we assess the validity of these assumptions by using traffic traces, collected from a deployed network of access points, in [36]. We model users who arrive and connect to AP by a homogeneous Poisson process with intensity λ_{Arr}. We denote the time users stay connected to AP by T_{Dur}, which follows an exponential distribution with average $\frac{1}{\lambda_{\text{Dur}}}$. We assume the system to be stationary with respect to user connections and disconnections. Based on Little's law [30], the average number of connected users at any time instant t is constant and given by: $N_{\text{Con}} = \lambda_{\text{Arr}}/\lambda_{\text{Dur}}$.

Users generate requests independently of each other. For each user, the three types of requests she makes are also independent: Standard and authenticated requests are modeled by independent homogeneous[3] Poisson processes with intensity λ_{Std} and λ_{Auth}, respectively. We assume that each user makes a request when she connects to AP. For instance, e-mail or RSS clients automatically connect to a server when an Internet connection is available. We assume that only a proportion α_{LBS} of the users make LBS requests, and we model such requests by independent homogeneous Poisson processes with intensity λ_{LBS} for each user.

Fig. 3 depicts the user arrivals, departures, standard and LBS request processes and illustrates the key notations and concepts introduced in this section.

4.2 Threat

We first focus on a single sub-interval and quantify the location-privacy threat, with respect to the number of users whose locations are disclosed to the adversary because of others. Specifically, we call a *victim* a user who makes a standard request at a time at which the adversary knows the (IP, Location) mapping.

Quantifying the Threat in a Sub-interval. If at least one user connected to AP uses an LBS at some time instant (thus revealing her current location), $\mathcal{A}$ obtains the (IP, Location) mapping based on which it can locate other users.

We define the *compromise time* T_{Comp} as the first time within the sub-interval, when a user connected to AP uses an LBS. If such an event does not occur, the compromise time is equal to T. At any time, there are on average

[3] We use homogeneous Poisson processes for simplicity. A model using inhomogeneous processes with piece-wise constant intensity is available as a technical report [36].

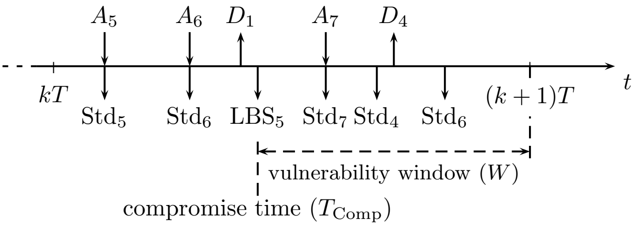


Fig. 3. Threat caused by a user making an LBS request. A_i and D_i represent User i's arrival and departure, respectively. Users 1 and 4 are already present at time kT. The time at which the first LBS request is made (LBS_5) is called the *compromise time* (T_{Comp}). From time T_{Comp} on, any user who makes a standard request is a victim. Users already connected at T_{Comp} are victims if they make a standard request after T_{Comp}, e.g., User 4. Users who connect after T_{Comp} are, *de facto*, victims as users make a standard request when they connect, e.g., User 7.

N_{Con} users connected to AP, out of which $\alpha_{\mathrm{LBS}} N_{\mathrm{Con}}$ potentially make LBS queries. The aggregated process of LBS requests is a Poisson process with intensity $\Lambda_{\mathrm{LBS}} = \alpha_{\mathrm{LBS}} N_{\mathrm{Con}} \lambda_{\mathrm{LBS}}$. Therefore, the expected compromise time is $\frac{1}{\Lambda_{\mathrm{LBS}}}(1 - e^{-\Lambda_{\mathrm{LBS}} T})$. We call $F_{\mathrm{Comp}}(\bar{t})$ the probability that at least one LBS query (from the aggregated process) is made before time $\bar{t}$ in the sub-interval and f_{Comp} the corresponding probability density function. The time interval that spans from the compromise time to the end of the sub-interval is called the *vulnerability window* (see Fig. 3) and the expected value of its duration W is

$$\mathbf{E}\left[W\right] = T - \frac{1 - e^{-\Lambda_{\mathrm{LBS}} T}}{\Lambda_{\mathrm{LBS}}}. \tag{1}$$

Fig. 4 depicts the cumulative distribution function of the compromise time and its average value in an example setting. We observe that even with moderate AP popularity and LBS usage, the adversary obtains the mapping before the DHCP lease expires in 83% of the cases and he does so after 11 hours on average.

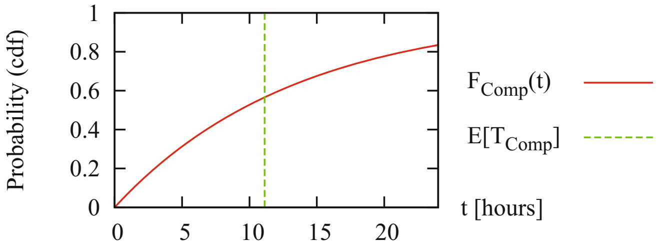


Fig. 4. Cumulative distribution function of the compromise time T_{Comp} (expressed in hours). The parameters were set to T = 24 h, $\lambda_{\mathrm{Arr}} = 5$ users/h, $\lambda_{\mathrm{Dur}} = 1/1.5$ (i.e., average connection time of one hour and a half), $\lambda_{\mathrm{LBS}} = 0.05$ req./h, and $\alpha_{\mathrm{LBS}} = 0.2$.

To compute the number of victims, we distinguish between two groups of users: those who were already connected when the first LBS request was made, e.g., User 6 in Fig. 3, and those who connected during the vulnerability window (and are, *de facto*, victims as they make a standard request when they connect), e.g., User 7. We call V the number of victims. It can be shown that (see [36]):

$$\mathbf{E}\left[V\right] = \frac{N_{\mathrm{Con}}\Lambda_{\mathrm{LBS}}\lambda_{\mathrm{Std}}}{(\lambda_{\mathrm{Std}}+\lambda_{\mathrm{Dur}})-\Lambda_{\mathrm{LBS}}}\cdot\left[\frac{1-e^{-\Lambda_{\mathrm{LBS}}T}}{\Lambda_{\mathrm{LBS}}}-\frac{1-e^{-(\lambda_{\mathrm{Std}}+\lambda_{\mathrm{Dur}})T}}{(\lambda_{\mathrm{Std}}+\lambda_{\mathrm{Dur}})}\right]+ \lambda_{\mathrm{Arr}}\left(T-\frac{1-e^{-\Lambda_{\mathrm{LBS}}T}}{\Lambda_{\mathrm{LBS}}}\right). \quad (2)$$

This number has to be compared to the average number of users who have been connected at some point within the sub-interval: $V_{\mathrm{tot}} = N_{\mathrm{Con}} + \lambda_{\mathrm{Arr}}T$. It can be seen in Fig. 5 that the proportion of victims $\mathbf{E}\left[V\right]/V_{\mathrm{tot}}$ increases with T. This is because all users who connect after the compromise time are victims. When the DHCP lease expires, the location of more than half of the users is compromised.

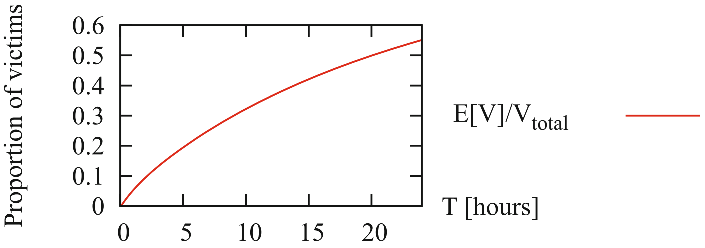
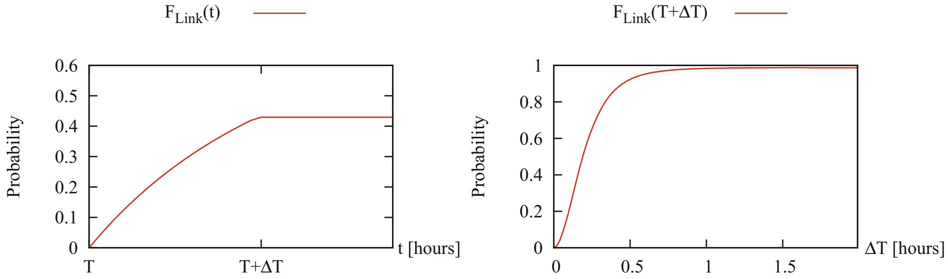


Fig. 5. Proportion of victims within a sub-interval of length T, corresponding to a DHCP lease. The parameters were set to: $\lambda_{\mathrm{Arr}} = 5$ users/h, $\lambda_{\mathrm{Dur}} = 1$ (i.e., average connection time of one hour), $\lambda_{\mathrm{Std}}=10$ req./h, $\lambda_{\mathrm{LBS}}=0.05$ req./h, and $\alpha_{\mathrm{LBS}}=0.2$.

Inferring IP Change. We consider two successive sub-intervals, without loss of generality I_0 and I_1, and we look at the linking probability F_{Link} that the adversary infers the IP change from authenticated requests. This occurs if at least one user makes both an authenticated request at most ΔT time units ($\Delta T < T/2$) before, and another authenticated request at most ΔT time units after, the IP change. An expression of the probability F_{Link} of inferring the IP change can be derived by distinguishing between the users who were connected at time $T - \Delta T$ and those who connected within $[T - \Delta T, T]$ (see [36]).

The linking probability can be thought of as depending both on t and ΔT. Fig. 6a depicts the linking probability as a function of t. It remains constant for $t \geq T + \Delta T$ because only authenticated requests made in the time interval $[T-\Delta T, T+\Delta T]$ are taken into account to infer the IP change. Note that with a value of ΔT as small as 5 minutes, which provides high confidence, the adversary can still infer the IP change with a probability of 43%.

Fig. 6b depicts the linking probability at time $T + \Delta T$ as a function of ΔT. It can be observed that this probability rapidly converges to 1. Note that the fact that linking probability increases with ΔT is balanced by the decreased confidence of the adversary. This is because the probability that a user makes two authenticated requests from two distinct access points in the time interval $[T - \Delta T, T + \Delta T]$ (moving from one to the other) increases with ΔT.



(a) Probability of inferring the IP change before time $t > T$ ($\Delta T = 5$ min)

(b) Linking probability at time $T + \Delta T$

Fig. 6. Linking probability. The parameters were set to $\lambda_{\text{Arr}} = 5$ users/h, $\lambda_{\text{Dur}} = 1/1.5$, $\lambda_{\text{Std}} = 10$ req./h, $\lambda_{\text{LBS}} = 0.05$ req./h, $\lambda_{\text{Auth}} = 2$ req./h, and $\alpha_{\text{LBS}} = 0.2$.

Quantifying the Threat over Multiple Sub-intervals. We now look at the probability (F_{Map}) of the adversary having the mapping, which is a combination of the probabilities that the compromise happens due to LBS usage (F_{Comp}) and the probability of having the mapping and inferring the IP change upon the lease expiration (F_{Link}), over successive sub-intervals. The probability $F^{(k)}_{\text{Map}}(t)$ that the adversary knows the mapping at time $t \in I_k$, $k \geq 1$ is

$$F^{(k)}_{\text{Map}}(\bar{t}) = F_{\text{Comp}}(\bar{t}) + (1 - F_{\text{Comp}}(\bar{t})) \cdot F^{(k-1)}_{\text{Map}}(T) \cdot ((1 - p_{\text{New}}) + p_{\text{New}} F_{\text{Link}}(\bar{t})) \quad (3)$$

with initial condition $F^{(0)}_{\text{Map}}(\bar{t}) = F_{\text{Comp}}(\bar{t})$. From Equation (3), it can be seen that $F^{(k)}_{\text{Map}}(T)$ obeys the following recursive equation:

$$F^{(k)}_{\text{Map}}(T) = a + bF^{(k-1)}_{\text{Map}}(T)$$

where $a = F_{\text{Comp}}(T)$ and $b = (1 - F_{\text{Comp}}(T)) \cdot ((1 - p_{\text{New}}) + p_{\text{New}} F_{\text{Link}}(T))$. This recursive equation has $a(1 - b^{k+1})/(1 - b)$ as a solution. As $b < 1$, $F^{(k)}_{\text{Map}}(T)$ converges to a finite value, i.e., $a/(1 - b)$.

The number of victims in the sub-interval I_k can be computed by replacing the density f_{Comp} with the density of $F^{(k)}_{\text{Map}}$ in the derivation of Equation (2) (see [36]). The probability that the adversary has the mapping (IP, Location) at time t in sub-interval I_k, i.e., $F^{(k)}_{\text{Map}}$ is illustrated in Fig. 7. It can be observed that the mapping probability increases over time and, after the convergence, the adversary successfully obtains the mapping before the DHCP lease expires in 79% of the cases and before the half-lease in 60% of the cases.

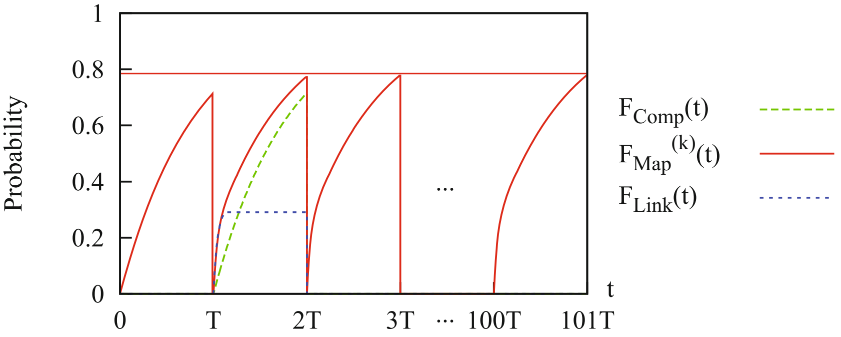


Fig. 7. Probability of knowing the (IP, Location) mapping at time t over several sub-intervals. The solid curve represents the probability of knowing the mapping at time t. The dashed curve represents the probability of obtaining the mapping from an LBS request. The dotted curve represents the probability of inferring the IP change. The parameters were set to $\lambda_{\text{Arr}} = 5$ users/h, $\lambda_{\text{Dur}} = 1/1.5$, $\lambda_{\text{Std}} = 10$ req./h, $\lambda_{\text{LBS}} = 0.035$ req./h, $\lambda_{\text{Auth}} = 0.2$ req./h, $T = 24$ h, $\Delta T = 2$ h, $\alpha_{\text{LBS}} = 0.1$, and $p_{\text{New}} = 1$. To highlight the respective contributions of the linking and compromise probabilities, some values differ from our previous setting (e.g., ΔT). In the first sub-interval, the linking probability is zero and the probability of having the mapping is the compromise probability. In subsequent sub-intervals, the probability $F^{(k)}_{\text{Map}}(t)$ increases due to the potential inference of IP changes: it becomes a combination of $F_{\text{Link}}(\bar{t})$ and $F_{\text{Comp}}(\bar{t})$ (and the probability of having the mapping by the end of the preceding sub-interval).

5 Experimental Results

In this section, we complement our theoretical analysis with experimental results based on traces from a network of Wi-Fi access points deployed at EPFL.

Dataset. Our dataset consists of daily user Wi-Fi *session traces*, *traffic traces* and *DNS traces* for a period of 23 days in June 2012. We aggregate the data of two APs (in a cafeteria and a library) located very close to each other (∼15 meters), to emulate a single popular hotspot and to avoid side effects of micro-mobility, i.e., devices frequently changing the AP they are connected to.

Session traces contain information related to users connecting and disconnecting from the APs, obtained from the RADIUS logs. There are three types of RADIUS events: (i) `start` – upon successful authentication the device is assigned an IP denoting the beginning of a session; (ii) `update` – a periodic status message; and (iii) `stop` – a user disconnects denoting the end of the session. Each log entry contains a timestamp, the device's anonymized MAC address (i.e., encrypted with a key that is changed daily), the assigned IP, the ID of the AP the device is connected to, and an event type.

Traffic traces are obtained from the logs at a border router connecting the network to the Internet. Each log entry contains a timestamp, the source IP, and the destination (including the IP and port). The mapping between a user's assigned IP and her MAC address allows to correlate traffic with session traces.

DNS traces are obtained from the local DNS servers. Each log entry contains a timestamp, the source IP and the requested host name. Based on the source IPs, timestamps and requested resources, we correlate DNS with traffic traces.

In total, 4,302 users connected to the AP during 23 days. Users typically begin arriving around 7:AM. The average number of users connected to the AP over a day (averaged over 23 days) increases during the day and peaks around 6:PM (136 users on average). Very few users are connected after midnight.

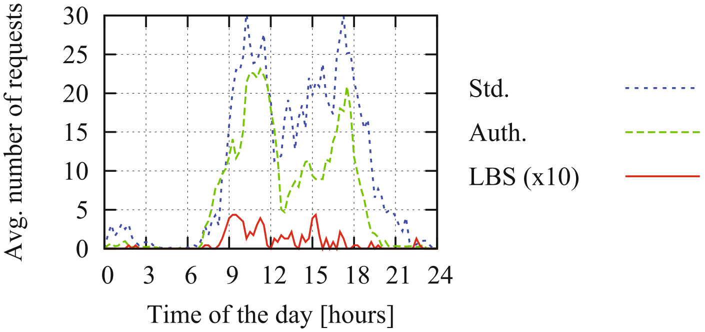


Fig. 8. Average number of std., auth. and LBS requests to the monitored services over a day (averaged over 23 days). For readability reasons, LBS traffic is multiplied by 10.

We filtered traffic to a number of Google services (including e-mail, search, LBS, analytics, advertising) and classified each request (i.e., standard, LBS, or authenticated) based on the destination IP, port and DNS requests. Details about the monitored services and the classification methodology can be found in citeRR. We sanitized the traffic data beforehand by appropriately grouping traffic traces into user-service sessions. To do so, we correlated traffic and DNS requests. This was possible because DNS replies for Google services are cached for a relatively short time (i.e., TTL of 300 seconds), and therefore a traffic request is very often preceded by a DNS request. Consequently, a request accounts for a user-service interaction, regardless of how much traffic the interaction generates.

Traffic to the monitored services (in terms of the number of user-service sessions) constitutes about 17% of the total traffic generated at the AP and 81.3% of users who connected have accessed at least one of the services. The average numbers of standard, authenticated and LBS requests (i.e., user-service interactions) during a day to the monitored services are depicted in Fig. 8. Standard requests are prevalent, followed by authenticated requests. The moderate usage of LBS can be explained with the location of the APs: most of the users visit this area almost on a daily-basis, therefore the need for location-based information is expected to be low. In our dataset, 9.5% of users generate LBS requests.

Results. First, we measure the compromise time and the proportion of victims based on the traces from our dataset. We compare the averaged experimental results with those from our theoretical analysis (Fig. 9). For the theoretical analysis, we use our framework with the parameters extracted from the real traces: $\lambda_{\text{Arr}} = 14.54$ users/h and an average connection time of 2.17 hours ($\lambda_{\text{Dur}} = 1/2.17$), obtained from the session traces; and traffic rates of $\lambda_{\text{Std}} = 28.3$ req./h, $\lambda_{\text{Auth}} = 14.6$ req./h and $\lambda_{\text{LBS}} = 0.16$ req./h (with $\alpha_{\text{LBS}} = 0.095$), obtained from the traffic traces. Because the theoretical model assumes a homogeneous user arrival rate, we compute the expected proportion of victims and compromise time as if the arrival process spanned from 7:30:AM–the time at which a significant number of users start connecting to the AP in our traces–to 7:PM. It can be observed that although the model does not capture the time-of-the-day effects of the user arrival and traffic processes, the theoretical and experimental expected proportions of victims match when considering the entire period of a day.

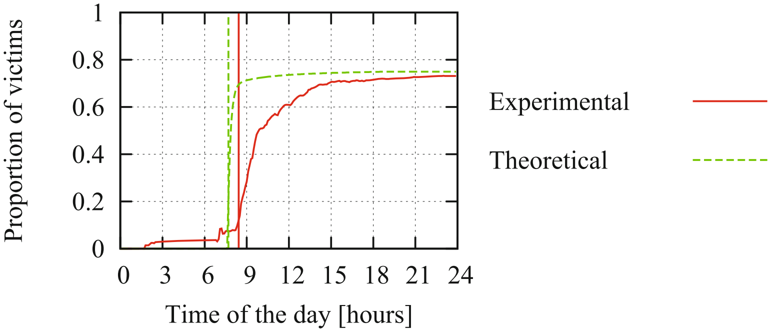


Fig. 9. Expected proportion of victims. Vertical lines represent average compromise times: theoretical T_{Comp} = 7:42:AM and experimental T_{Comp} = 8:25:AM.

We observe that around 8:AM (7:42AM estimated with our theoretical analysis and 8:25AM with our experimental results), only 1 hour after users typically start connecting to the AP, users' location privacy is compromised. By the end of the day, about 73% of the users who connected through the AP were compromised, out of which 90.5% did not make any LBS request ($\alpha_{\text{LBS}} = 0.095$). With respect to the number of users who use Google services the proportion of victims actually corresponds to 90%. Thus, the result shows that Google is able to learn the location of 90% of its users who connect from the AP.

Once the adversary obtains the (IP, Location) mapping, it can maintain it over time by relying on authenticated requests to infer the IP changes upon DHCP lease expirations, as discussed in Section 4. Using traces from our dataset, we compute the probability of the adversary inferring the IP change for different renewal times during a day, considering the authenticated requests made at most ΔT minutes before and after the IP is changed. We consider three different values, $\Delta T = 1$, $\Delta T = 5$ and $\Delta T = 10$ minutes, and show the results in Fig. 10. Even with the smallest inference time window of 1 minute, the adversary can infer the IP change with the probability 1 between 2:PM and 5:PM. With higher values of ΔT the time during which the adversary can infer with probability 1 is even longer, i.e., from 11:AM to 7:PM with $\Delta T = 10$. However, the adversary's confidence decreases with larger ΔT. During the periods with less traffic (e.g., from 11:PM to 6:AM), the probability of the adversary inferring the mapping is

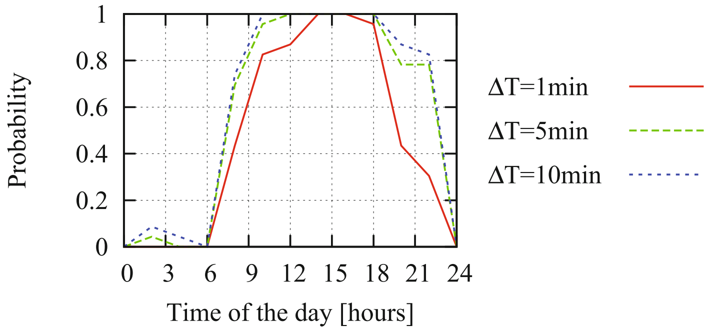


Fig. 10. Linking probability (i.e., probability of inferring the IP change) as a function of the renewal time, for different inference time window lengths (ΔT).

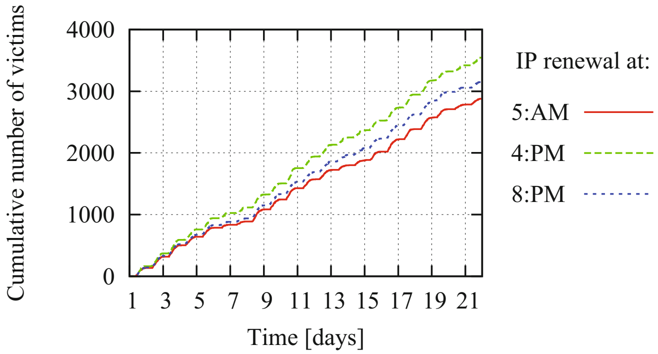


Fig. 11. Cumulative number of victims at AP during the whole experiment, for three different IP renewal times (the IP is changed every time the lease expires, $p_{\mathrm{New}} = 1$).

smaller (less than 0.2) in all the cases. Between 5:AM and 6:AM, the adversary cannot infer the IP change, as there is no traffic during this time.

To further confirm the importance of the IP renewal time and its effect on the adversary's success, we plot the cumulative number of victims compromised at AP during three weeks, depending on the IP renewal time (Fig. 11). We set $\Delta T =$ 5 minutes and we consider the renewal times at 5:AM, 4:PM and 8:PM, when the adversary is expected to be least successful, most successful and moderately successful, respectively. Indeed, from the results in Fig. 11, we confirm that the highest number of users (3,545 out of 4,302 users) is compromised when the IP renewal is at 4:PM, followed by 8:PM (3,149 victims). The adversary is least successful when the IP renewal is at 5:AM (compromising 2,879 users).

6 Countermeasures

Cryptographic primitives are efficient at protecting users' privacy, but because of the way networking protocols operate, they might not be sufficient, in particular, when the private information is the source IP address.

Hiding users' actual source IPs from the destination (i.e., the adversary) is a straightforward countermeasure against the considered threat and can be done in several ways. In relay-based anonymous communications, a user's traffic is forwarded from the source to the destination by several relay nodes, in a way that the destination cannot know the user's source IP. Examples of such networks include Tor [8], mix networks [5,7], or simple HTTP proxies. With Virtual Private Networks (VPNs), the user is assigned an IP that belongs to a remote network (e.g., a corporate network or commercial/public VPN). To the adversary, the user's requests appear to originate from within the remote network, whose location is different from that of the user. Unfortunately, such techniques are not widely adopted, especially in the case of mobile communications [34]. In addition, several techniques exist to identify the source IP of a client, even behind a NAT or a proxy, e.g., by using a Java applet [25].

Alternatively, these countermeasures can be implemented by ISPs, for instance, by deploying a country-wide NAT that aggregates traffic from all their subscribers at several gateways (e.g., Telefonica [33], Swisscom Hotspots) or by IP Mixing [29]. This also applies to operators of AP networks (e.g., Starbucks, AT&T Wi-Fi). However, they might lack incentives to implement such solutions.

Another approach to thwart the threat consists in degrading the knowledge of the adversary, by reducing the accuracy of the reported location and by increasing the uncertainty about the AP's location. Examples of location privacy enhancing technologies (PETs) reducing adversary's accuracy include spatial cloaking [2, 16] and adding noise to reported locations [1]. To increase adversary's uncertainty, [22] proposes to inject "dummy" requests, i.e., not related to users' locations. It is not easy for users to deploy these PETs, because some geolocation requests are implemented in operating systems, that can be controlled by the adversary (e.g., Google Android). Moreover, when these PETs are implemented in a non-coordinated fashion, the adversary might still be able to infer the actual location by filtering out requests that stand out from the bulk (increasing its certainty) and averaging the remaining requests (increasing its accuracy). Better results might be achieved if the AP operators implement the location-privacy preserving mechanisms, but they might lack incentives to do so.

Finally, as highlighted by our analysis, various other countermeasures can be implemented by the ISP or the AP's owner: reduce the DHCP lease, always allocate a new IP, trigger the IP change when the traffic is low (e.g., at 5:AM as suggested by our experimental results) or purposely impose silent periods around the renewal time (reducing the probability that the adversary infers the IP change from authenticated requests). Unfortunately, all these techniques have a negative effect on the quality of service and impose a significant overhead in network management. Thus, they are unlikely to be deployed in practice. Besides technical countermeasures, we envision a "Do-not-geolocalize" initiative, similar to "Do-not-track" [9], letting users to opt-out of being localized.

7 Discussion

Scale and Implications of the Threat. The threat enables an adversary to build an IP-location system, to obtain (at least) sporadic user locations and to profit from delivering location-targeted information when users access the services. However, we can also envision a different type of adversary, whose goal is to mount more powerful attacks on user privacy. In fact, once the adversary has access to sporadic user location, he is able to reconstruct entire trajectories, produce patterns of user-movement habits, or infer other information about the user, e.g., users' real identities, interests and activities. For example, in [31] it is shown how an adversary that observes each user's sporadic locations (that could be noisy and anonymized) can de-anonymize the users, compute the probability that a given user is at a given location at a given time, and construct users' full trajectories. By using various techniques, it has been shown that users can be identified by inferring where they spend most of their time (notably their home and workplace) [3,12,18,23]. In these cases, the identified location-privacy threat can serve as a building block that enables other, more powerful attacks.

In this paper, we focus on how an adversary can obtain the sporadic user-location information that is needed for commercial needs of service providers. Other attacks that are enabled by this location-privacy threat are beyond the scope of this paper and are largely addressed by the research community, as previously discussed. However, our work provides a framework that can be used to quantify sporadic location exposure upon which the community can build.

Business Opportunities. The presented (IP, Location) mapping technique can be used as a novel IP-location solution potentially improving on existing solutions [27,37]. Service providers, such as Google, can build and monetize this service by simply utilizing user traffic they receive. Additional advantages of this approach are that it does not require a dedicated infrastructure or network measurements. Such a system can be used on its own, or as a complementary to the existing ones. Because ISPs control the IP allocation and can prevent service providers from building the mapping (using the aforementioned countermeasure) they can make a profit by selling IP locations to service providers (e.g., Verizon in the US [6]) – some ISPs sell geographic information on the topology of their networks [25] – or by selling privacy-protection services to users.

8 Conclusion

In this paper we have presented a practical threat, demonstrating that the location privacy of users connecting to access points can be compromised by others. The scale of the threat is significant because it leverages on the way most networks are designed (i.e., NAT). When successful, the service provider can locate users within a few hundreds of meters, i.e., more accurately than existing IP-location databases. Our theoretical analysis provides a framework that enables us to quantify the threat for any access-point setting and to identify the key parameters and their impact on the adversary's success. The framework serves as a light-weight alternative to an extensive traffic analysis to estimate the threat. We experimentally investigate the state in practice, by analyzing real traces of users accessing Google services, collected from deployed Wi-Fi access points. We observe the large scale of the threat even with a modest use of LBS services. We survey possible countermeasures and we find that adequate ones can be used to protect individual users' location privacy, but they need to be widely deployed.

We intend to further study this threat by focusing on the following aspects: (i) the accuracy of a IP-location service, based on (IP, Location) mappings; (ii) the refinement of the model by modeling users' arrivals by an inhomogeneous Poisson process to capture time-of-the-day effects; (iii) the adversary's inference of IP changes, studying the trade-off between the probability of inferring the IP change and the adversary's confidence; and (iv) the adversary's ability to track users as they move and connect to different APs over time.

References

1. Agrawal, R., Srikant, R.: Privacy-Preserving Data Mining. In: SIGMOD (2000)
2. Ardagna, C.A., Cremonini, M., De Capitani di Vimercati, S., Samarati, P.: An Obfuscation-Based Approach for Protecting Location Privacy. IEEE Transactions on Dependable Secure Computing 8(1), 13–27 (2011)
3. Beresford, A., Stajano, F.: Location Privacy in Pervasive Computing. IEEE Perv. Comp. 2, 46–55 (2003)
4. Casado, M., Freedman, M.J.: Peering Through the Shroud: The Effect of Edge Opacity on IP-Based Client Identification. In: NSDI (2007)
5. Chaum, D.L.: Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms. Communications of the ACM 24(2), 84–90 (1981)
6. CNN: Your Phone Company is Selling Your Personal Data (2011), http://money.cnn.com/2011/11/01/technology/verizon_att_sprint_tmobile_privacy
7. Danezis, G., Dingledine, R., Hopwood, D., Mathewson, N.: Mixminion: Design of a Type III Anonymous Remailer Protocol. In: S&P, pp. 2–15 (2003)
8. Dingledine, R., Mathewson, N., Syverson, P.: Tor: The Second-generation Onion Router. In: USENIX Security (2004)
9. Federal Trade Commission: Protecting Consumer Privacy in an Era of Rapid Change: A Proposed Framework for Businesses and Policymakers. Report (2010)
10. Freedman, M.J., Vutukuru, M., Feamster, N., Balakrishnan, H.: Geographic Locality of IP Prefixes. In: IMC (2005)
11. Ghosh, A., Jana, R., Ramaswami, V., Rowland, J., Shankaranarayanan, N.: Modeling and Characterization of Large-Scale Wi-Fi Traffic in Public Hot-Spots. In: INFOCOM (2011)
12. Golle, P., Partridge, K.: On the Anonymity of Home/Work Location Pairs. In: Tokuda, H., Beigl, M., Friday, A., Brush, A.J.B., Tobe, Y. (eds.) Pervasive 2009. LNCS, vol. 5538, pp. 390–397. Springer, Heidelberg (2009)
13. Goodell, G., Syverson, P.: The right place at the right time. Communications of the ACM 50(5), 113–117 (2007)
14. Google Engineering Center Zurich: Technology and Innovation for Web Search. Private communication (October 2012)
15. Google Privacy Policy (2012), http://www.google.com/intl/en/policies/privacy/

16. Gruteser, M., Grunwald, D.: Anonymous Usage of Location-Based Services Through Spatial and Temporal Cloaking. In: MobiSys (2003)
17. Guo, C., Liu, Y., Shen, W., Wang, H., Yu, Q., Zhang, Y.: Mining the Web and the Internet for Accurate IP Address Geolocations. In: INFOCOM (2009)
18. Hoh, B., Gruteser, M., Xiong, H., Alrabady, A.: Enhancing Security and Privacy in Traffic-Monitoring Systems. IEEE Perv. Comp. 5, 38–46 (2006)
19. HostIP: My IP Address Lookup and Geotargeting Community Geotarget IP Project, `http://www.hostip.info/`
20. Targeting Local Markets: An IAB Interactive Advertising Guide. Interactive Advertising Bureau (2010)
21. Katz-Bassett, E., John, J.P., Krishnamurthy, A., Wetherall, D., Anderson, T., Chawathe, Y.: Towards IP Geolocation Using Delay and Topology Measurements. In: IMC (2006)
22. Kido, H., Yanagisawa, Y., Satoh, T.: An Anonymous Communication Technique using Dummies for Location-Based Services. In: ICPS, pp. 88–97 (2005)
23. Krumm, J.: Inference Attacks on Location Tracks. In: LaMarca, A., Langheinrich, M., Truong, K.N. (eds.) Pervasive 2007. LNCS, vol. 4480, pp. 127–143. Springer, Heidelberg (2007)
24. Geolocation and online fraud prevention by MaxMind, `http://www.maxmind.com/`
25. Muir, J.A., Oorschot, P.C.V.: Internet Geolocation: Evasion and Counterevasion. ACM Computing Survey 42, 4:1–4:23 (2009)
26. Patil, S., Norcie, G., Kapadia, A., Lee, A.: "Check Out Where I Am!": Location-Sharing Motivations, Preferences, and Practices. In: CHI (2012)
27. Poese, I., Uhlig, S., Kaafar, M.A., Donnet, B., Gueye, B.: IP Geolocation Databases: Unreliable? ACM SIGCOMM CCR 41, 53–56 (2011)
28. PricewaterhouseCoopers: Internet Advertising Revenue Report (2011)
29. Raghavan, B., Kohno, T., Snoeren, A.C., Wetherall, D.: Enlisting ISPs to Improve Online Privacy: IP Address Mixing by Default. In: Goldberg, I., Atallah, M.J. (eds.) PETS 2009. LNCS, vol. 5672, pp. 143–163. Springer, Heidelberg (2009)
30. Ross, S.M.: Stochastic Processes. Wiley (1995)
31. Shokri, R., Theodorakopoulos, G., Le Boudec, J.Y., Hubaux, J.P.: Quantifying Location Privacy. In: S&P (2011)
32. Skyhook Location Perf, `http://www.skyhookwireless.com/location-technology`
33. Telefonica implement NAT for DSL users (2012), `http://bandaancha.eu/articulo/7844/usuarios-adsl-movistar/compartiran-misma-ip-mediante-nat-escasear-ipv4`
34. Tor Metrics Portal, `https://metrics.torproject.org`
35. USA Department of Defenses: Global Positioning System: Standard Positioning Service Performance Standard (2008)
36. Vratonjic, N., Huguenin, K., Bindschaedler, V., Dubovitskaya, A., Hubaux, J.P.: Location Privacy Threats at Public Hotspots. Tech. rep., EPFL (2013)
37. Wang, Y., Burgener, D., Flores, M., Kuzmanovic, A., Huang, C.: Towards Street-Level Client-Independent IP Geolocation. In: NSDI (2011)
38. Xie, Y., Yu, F., Achan, K., Gillum, E., Goldszmidt, M., Wobber, T.: How Dynamic are IP Addresses? In: SIGCOMM (2007)
39. Yen, T.F., Xie, Y., Yu, F., Yu, R.P., Abadi, M.: Host Fingerprinting and Tracking on the Web: Privacy and Security Implications. In: NDSS (2012)