A Location-Privacy Threat Stemming from the Use of Shared Public IP Addresses

Nevena Vratonjic, *Student Member, IEEE*, Kévin Huguenin, *Member, IEEE*,
Vincent Bindschaedler, *Student Member, IEEE*, Jean-Pierre Hubaux, *Fellow, IEEE*

Abstract—This paper presents a concrete and widespread example of situation where a user's location privacy is *unintentionally* compromised *by others*, specifically the location-privacy threat that exists at access points (public hotspots, FON, home routers, *etc.*) that have a single public IP and make use of network address translation (NAT). As users connected to the same hotspot share a unique public IP address, a single user's making a location-based request is enough to enable a service provider to map the IP address 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 location-privacy threat caused by shared public IPs in combination with NAT. (ii) We formalize and analyze the threat theoretically. In particular we derive and provide expressions of the probability that the service provider will learn the mapping and of the expected proportion of victims. (iii) We experimentally assess the state in practice by using real traces (collected from deployed hotspots over a period of 23 days) of users who accessed Google services. We also discuss how existing countermeasures can thwart the threat.

Index Terms—Location Privacy; Network Address Translation (NAT); IP-Geolocation

1 Introduction

As mobile users benefit from online services while on the go, location information has become mainstream. On the one hand, by providing their location with so-called location-based services (LBS), users can enjoy context-aware features, such as finding nearby restaurants, and social features, such as sharing location information with their friends on social networks [2]. Although very convenient, the usage of LBS raises serious privacy issues, because much sensitive information can be inferred from users' locations, including users' movements and associated activities. On the other hand, location information has become essential for many online service providers [3], especially for those whose business models revolve around personalized services. A prominent example is (mobile) online advertising, an ever-increasing business whose worldwide annual revenue is in the tens of billions of US Dollars [4], as location-specific ads have become significantly more appealing to users [5].

Even if users do not willingly disclose their location to the service provider (typically a non-LBS service provider), the service can obtain users' locations through *IP-location*, i.e., determining the location of a device from its IP. Existing IP-location services rely either on (i) active techniques, typically based on network measurements [6], [7], or (ii) passive techniques, relying on databases with records of IP-location mappings [8], [9]. Active techniques provide more accurate results than passive ones, however, they incur high measurement overhead and a high response time (in the range of several seconds to several minutes). A passive approach is several orders of magnitude faster and, hence preferred by service operators. A number of IP-location databases are available, either for free (e.g., HostIP [9]) or commercial (e.g., MaxMind [8]). They provide a country-level accuracy, and at most city-level, and most of the entries refer only to a few countries [10]. For instance, MaxMind reports it correctly geo-locates, within a radius of 40 km, 81% of IP addresses in the US and 60%-80% in Europe. This level of accuracy is only effective for regional advertising but is not sufficient for local businesses (e.g., coffee shops) that require neighborhood or street-level accuracy [5]. Therefore, major web companies, including Google, are working on improving IP-location[1] by constructing their own IP-location databases. An

- *This article is a revised and extended version of a paper that appears in the Proceedings of the 13th Privacy Enhancing Technologies Symposium (PETS 2013) Vratonjic* et al. *[1].*
- *Nevena Vratonjic is with Kudelski Security, Cheseaux-sur-Lausanne, Switzerland (e-mail: nevena.vratonjic@nagra.com). This work was carried out while N. Vratonjic was with EPFL, Lausanne, Switzerland*
- *K. Huguenin and J.-P. Hubaux are with EPFL, Lausanne, Switzerland (e-mail: kevin.huguenin@epfl.ch; jean-pierre.hubaux@epfl.ch).*
- *V. Bindschaedler is with the University of Illinois at Urbana-Champaign (UIUC), Urbana-Champaign, IL 61801, USA (e-mail: bindsch2@illinois.edu). This work was carried out while V. Bindschaedler was with EPFL, Lausanne, Switzerland.*

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 [11].

original way for the service providers to obtain a user's location is via transitivity by relying on other users to disclose their location (typically LBS users) and that of others in their vicinity (typically in a social network): if a provider knows the location of user B and that user A is close to user B, the provider knows roughly the location of A. For instance when users *check-in* on online social networks and tag friends who are with them. Even if the proximity information is not directly revealed by users, the adversary is still able to infer this, as we will show in the case of access points that make use of network address translation.

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 (or made) LBS queries, the service provider (the adversary in this paper) learns the user's location. Indeed, because the devices connected to a public hotspot, implementing network address translation, share the AP's public IP address, when users generate LBS queries, the service provider learns the fine-grained geographic location of the AP and maps it to the AP's public IP. IP addresses remain the same for a certain amount of time, therefore 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.

Our contribution in this paper is three-fold: (i) We identify the location-privacy threat that arises from the use of shared IPs. (ii) We formalize and analyze the problem and we provide a framework for estimating 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 that can be extracted from logs), and it quantifies the potential threat. It is a light-weight alternative to extensive traffic analysis. (iii) We evaluate experimentally the scale of the threat using real traces (collected for a period of one month from deployed hotspots) of users who accessed Google services. Even at a moderately often 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 first connect and within 24 hours it can locate up to 73% of the users. 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: related works either focus on *other* attacks to infer a user's location or on *how* sporadic location exposure can be exploited to track a user or to infer high-level information about her.

Our paper is organized as follows. In Sec. 2 we provide the relevant background. We describe the system setting, the adversary and the threat model in Sec. 3. We formalize the problem by modeling user behaviors in Sec. 4 and we analytically quantify the threat by providing closed-form expressions of the number of victims. We further evaluate the threat based on traces from deployed access points and present the results in Sec. 5. In Sec. 6, we consider possible countermeasures. Further discussion is presented in Sec. 7. Finally, we conclude in Sec. 8.

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* or *dynamic*, i.e., periodically obtained from a pool of available addresses, typically through the Dynamic Host Configuration Protocol (DHCP). The host can use the IP 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 the period of one month, less than 1% of the clients used more than one IP and less than 0.07% of clients used more than three IP addresses [12].

Network Address Translation (NAT). In order to cope with IP address depletion, NAT was introduced. NAT hides an 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 also 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 [12].

Geolocation. Mobile devices determine their positions by using their embedded GPS or an online geolocation service. With a GPS unit, the computation takes place locally by using satellites positions. Commercial GPS units provide highly accurate location results (less than ten meters) [13], especially in open sky environments. With online geolocation services (e.g., Skyhook) a device shares the list of nearby cell towers and Wi-Fi APs, together with their signal strengths, based on which the server estimates the device location by using a reference database. This database is built typically by deploying GPS-equipped mobile units that scan for cell towers and Wi-Fi APs and plot

their precise geographic locations. In addition, they take into account input reported by users with GPS-equipped devices who provide both their positions and the surrounding parameters. The accuracy of such systems is in the range of 10 meters [14].

Note that Skyhook cannot be used by a service provider to infer users' locations from their IP. Indeed, Skyhook provides only APs' MAC addresses to location mappings and the service provider does not know the MAC addresses of a user's neighboring APs (not even the MAC address of the AP the user connects from) unless the user discloses them.

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 or a home (wireless) router, as depicted in Fig. 1. An AP, located at (x_1,y_1), is connected to the Internet and provides connectivity to the users. The AP has a single *dynamic public* IP address that is allocated with DHCP by its Internet provider (from a given pool of available IPs) and that is valid during the DHCP lease time. The AP operator has no control over the way the AP's IP is assigned by the Internet provider. The AP implements NAT.

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 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));
3) Authenticated standard (i.e., that are neither Geolocation nor LBS) 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 position in free-text (e.g., "bars near Park and 57th, Manhattan") or by pin-pointing her position on a map. The location can also be determined by the user's device (see Sec. 2) and communicated to the service provider by a mobile app or by her browser through the `geolocation.getCurrentPosition` JS function used by websites. 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 refer to both as LBS requests. For all types of request, the server knows the source IP, i.e., the AP's public IP.

3.2 Adversary and Threat Models

We consider an adversary whose goal is to learn users' current (or past) 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 collected by a number of servers that provide services described above. Companies, such as Google, provide web searches (Google), e-mail (GMail), social networking (Google+), and geolocation and location-based services (Google Maps). As such, Google receives requests of the three types and consolidates all the information obtained [15]. The extent to which these services are used is exacerbated by their deep integration in the widely spread Android operating system, which frequently sends the devices' locations to Google services[3]. 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 working on improving its IP-location based on users' traffic, by mining location-related events (e.g., search queries associated with location such as "best burgers Manhattan") [11].

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 provider to receive all three types of requests and a significant fraction of user traffic. Instead, they can do so through aggregation.

2. We assume that all LBS requests concern users' actual locations, or that the server has means to distinguish between such LBS requests and other LBS requests. This is the case when the location is obtained directly using the methods described in Sec. 2.

3. Note that mobile phones are less susceptible to the threat presented in this paper as they often access the Internet over cellular networks (e.g., 3G/4G) instead of Wi-Fi access points.

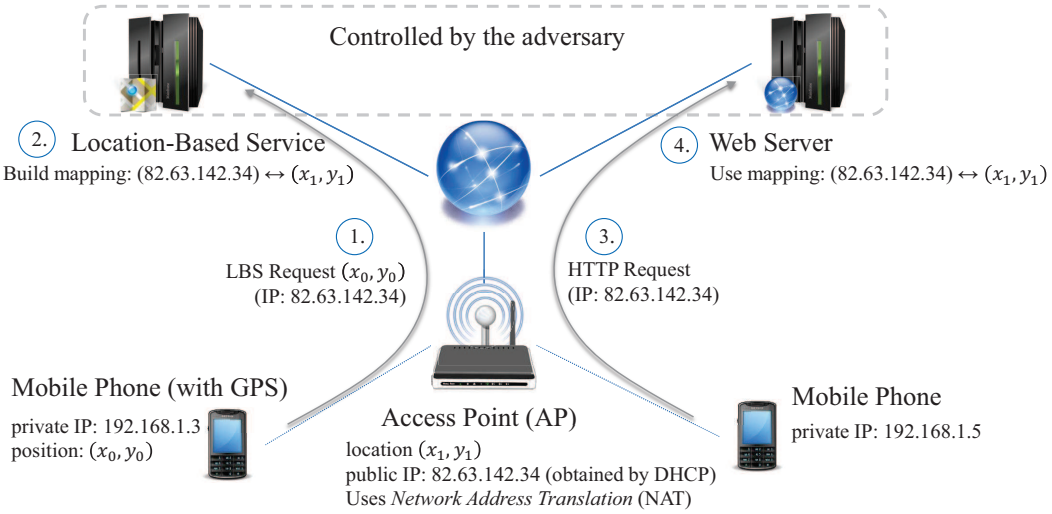

Fig. 1. System and threat model. Devices connect through a NAT Access Point. A user who makes an LBS request reveals her location to the adversary (1) who maps the IP to its location (2). When another user connects to a different server (3), the adversary uses the mapping to locate her as she connects with the same IP (4).

In this paper, we focus on the case where the adversary has access to all three types of requests. The adversary is assumed to be *honest-but-curious*, meaning that he passively collects information.

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 the APs' IPs and their geographic coordinates based on LBS requests he receives from other users connected to the APs. 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 (assumed to be 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 accuracy of the GPS 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 can exploit the obtained mapping and infer from the source IP (i.e., the AP's public IP again) 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 specific users, he can locate those who make an authenticated request.

We assume that the IP addresses in the DHCP pool can be assigned to clients at very distant locations [16]. 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 the adversary 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 another request authenticated by the same cookie `john@dom.com` but originating from a different 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 time is short enough, then the adversary can infer, with high confidence, that the IP has changed and infer its new value.

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 in terms of 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

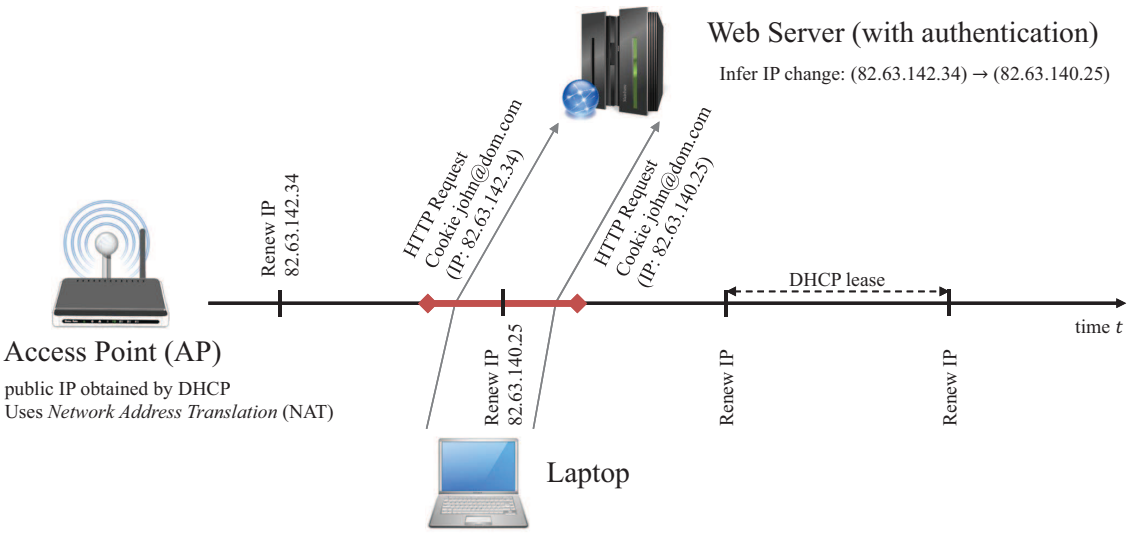

Fig. 2. AP's IP renewal and update of the (IP, Location) mapping. A user makes an authenticated request during a lease in which the adversary learns the mapping, shortly before and after the AP is assigned a new IP. The adversary infers that the AP's IP changed from `82.63.142.34` to `82.63.140.25` and updates the mapping.

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. Our experimental evaluation (see Sec. 5), based on a real dataset, shows that the threat is real: in a typical setting, Google learns the location of up to 90% of its users without them explicitly disclosing it.

4 Formalization and Analysis

In this section, we model the aforementioned setting and build a framework to quantify the threat, which takes only a few parameters as input. The notations are summarized in Table 1 in the appendix.

4.1 Model

We consider an access point *AP*, a passive 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 with 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: $I_k = [kT, (k+1)T]$, and we conduct a continuous-time analysis of the system within the sub-intervals. 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 [17] – 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., [18]). We model users who arrive and connect to *AP* with a homogeneous Poisson process with intensity λ_{Arr}, thus the number $N_{\text{Arr}}(t)$ of users who connect to *AP*, during any time interval of length t, follows a Poisson distribution with parameter $\lambda_{\text{Arr}}t$:

$$\mathbf{P}\left[N_{\text{Arr}}(t) = n\right] = \frac{(\lambda_{\text{Arr}}t)^n}{n!} e^{-\lambda_{\text{Arr}}t}\ ,\ n \geq 0.$$

We denote the time users stay connected to *AP* by T_{Dur}, which follows an exponential distribution with average $\frac{1}{\lambda_{\text{Dur}}}$. This means that the associated cumulative distribution function (cdf) and probability density function (pdf) are

$$f_{\text{Dur}}(t) = \lambda_{\text{Dur}} e^{-\lambda_{\text{Dur}}t}$$

and

$$F_{\text{Dur}}(t) = \mathbf{P}\left[T_{\text{Dur}} < t\right] = 1 - e^{-\lambda_{\text{Dur}}t}\ .$$

A noteworthy property of exponential distributions is *memorylessness*: the probability distribution of the time spent by a given user at a certain AP *since* a given

time instant t, provided that the user is still connected at time t, is the same for all t. In other words, $\forall t, \forall \delta t,\ \mathbf{P}\left[T_{\mathrm{Dur}} > \delta t\right] = \mathbf{P}\left[T_{\mathrm{Dur}} > t + \delta t \mid T_{\mathrm{Dur}} > t\right]$.

We assume the system to be stationary in terms of user connections and disconnections. Based on Little's law [17], the average number of connected users at any time instant t is therefore constant and given by: $N_{\mathrm{Con}} = \lambda_{\mathrm{Arr}}/\lambda_{\mathrm{Dur}}$.

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 Poisson processes with average intensity λ_{Std} and λ_{Auth}, respectively.[4] The probability that at least one request of a type is made during an interval of length t is

$$P_{\mathrm{Std}}(t) = 1 - e^{-\lambda_{\mathrm{Std}} t} \quad \text{and} \quad P_{\mathrm{Auth}}(t) = 1 - e^{-\lambda_{\mathrm{Auth}} t}\ .$$

Another noteworthy property of Poisson processes is that the numbers of requests in two disjoint intervals are independent. We assume that each user makes a request when she connects to *AP*. For instance, an e-mail or RSS client usually automatically connects 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, and the standard and LBS request processes and it illustrates the key notations and concepts introduced in this section.

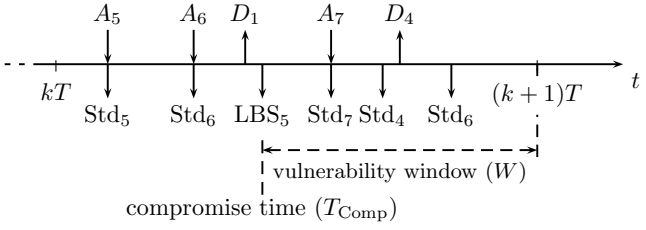

Fig. 3. Threat caused by a user making an LBS request. A_i and D_i represent User i's arrival and departure, respectively. Users 1 and 4 are already present at time kT. The time at which the first LBS request is made (LBS_5) is called the *compromise time* (T_{Comp}). From time T_{Comp} on, any user 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.

4. For the sake of clarity, the derivations and expressions presented in the paper are for homogeneous processes. We did the derivations for inhomogeneous Poisson processes with piece-wise constant (over the intervals 1:30AM - 7:30PM - 1:30PM - 7:30PM) intensities as well, and we included the corresponding graphs.

4.2 Threat

We first focus on a single sub-interval and quantify the location-privacy threat, in terms of the number of users whose locations are disclosed to the adversary because of other users. Specifically, we call a *victim* a user who makes a standard request at some point in time at which the adversary already knows the (IP, Location) mapping. Such a user is considered a victim "once-for-ever" (for the entire sub-interval).

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 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 probability that at least one LBS request (from the aggregated process) is made before time $\bar{t}$ is

$$F_{\mathrm{Comp}}(\bar{t}) = \mathbf{P}\left[T_{\mathrm{Comp}} < \bar{t}\right] = 1 - e^{-\Lambda_{\mathrm{LBS}} \bar{t}}\ ,$$

and the expected compromise time is $\frac{1}{\Lambda_{\mathrm{LBS}}}(1 - e^{-\Lambda_{\mathrm{LBS}} T})$. We call 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 W of its duration 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[5]. 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.

In order to compute the cumulative number of victims (all the users who made a standard request at some point in time in the vulnerability window), 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 subsequently 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_1 and V_2 the number of victims in each group.

There are N_{Con} users connected at the compromise time (recall that there are on average N_{Con} users

5. Throughout this section, we use different, yet reasonable, values for the parameters of the model in order to highlight singular aspects of the threat, e.g., the sensitivity of the different metrics and the respective contributions of the different parameters.

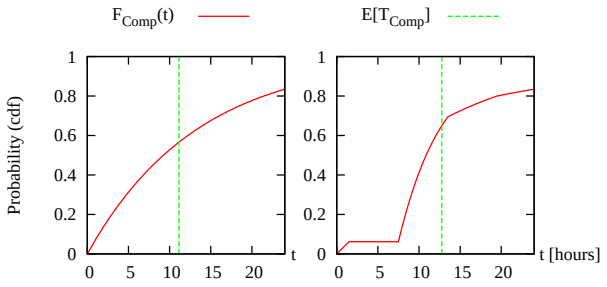

Fig. 4. Cumulative distribution function of the compromise time T_{Comp}. The parameters were set to $\lambda_{\text{Arr}} = 5$ users/h, $\lambda_{\text{Dur}} = 1/1.5$ (i.e., average connection time of one hour and a half), $\lambda_{\text{LBS}} = 0.05$ req./h, and $\alpha_{\text{LBS}} = 0.2$. Homogeneous (left) / inhomogeneous (right) Poisson processes for traffic.

in the system at any time). Whenever we compute an expected value involving the number of users connected, Wald's equation [17] allows us to consider that the system is composed of exactly N_{Con} users. Provided that an LBS request is made, the number of victims at that time is the number V_1 of connected users who make a standard request before leaving and before the end of the sub-interval. We compute the expected value of V_1 by applying the law of total probability, conditioning over both the compromise time and the time spent in the system. A user connects at time $t \in [0, T]$ (with a distribution f_{Comp}) and stays connected for a time $u \in [0, +\infty)$ (with a distribution f_{Dur}). The user is a victim if she makes a standard request before she disconnects and before the end of the lease, that is in the interval $[t, \min(T, t+u)]$; this happens with a probability P_{Std} that depends on the duration of the interval, i.e., $\min(u, T-t)$.

$$\begin{aligned}\mathbf{E}[V_1] &= N_{\text{Con}} \int_{t=0}^{T} \int_{u=0}^{\infty} f_{\text{Comp}}(t) f_{\text{Dur}}(u) P_{\text{Std}}(\min(u, T-t)) \\ &= N_{\text{Con}} \frac{\Lambda_{\text{LBS}} \lambda_{\text{Std}}}{(\lambda_{\text{Std}} + \lambda_{\text{Dur}}) - \Lambda_{\text{LBS}}} \cdot \\ &\qquad \left[\frac{1 - e^{-\Lambda_{\text{LBS}} T}}{\Lambda_{\text{LBS}}} - \frac{1 - e^{-(\lambda_{\text{Std}} + \lambda_{\text{Dur}})T}}{(\lambda_{\text{Std}} + \lambda_{\text{Dur}})} \right] \quad (2)\end{aligned}$$

The average number V_2 of users who connect to AP between the compromise time and the end of the sub-interval is:

$$\mathbf{E}[V_2] = \lambda_{\text{Arr}} \cdot \mathbf{E}[W] = \lambda_{\text{Arr}} \left(T - \frac{1 - e^{-\Lambda_{\text{LBS}} T}}{\Lambda_{\text{LBS}}} \right) . \quad (3)$$

The average number of victims in a sub-interval is the expectation of the sum of the number of victims connected at the compromised time (V_1) and victims arriving within the vulnerability window (V_2). This number has to be compared to the average number of users who have been connected at some point within the sub-interval: $V_{\text{total}} = N_{\text{Con}} + \lambda_{\text{Arr}} T$. It can be observed in Fig. 5 that the proportion of victims $(\mathbf{E}[V_1] + \mathbf{E}[V_2])/V_{\text{total}}$ increases with T. This is because all users who connect during the vulnerability window are victims. As the probability of the adversary obtaining the mapping before time T increases with T, V_1/V_{total} first increases. However, because V_1 is upper-bounded by N_{Con} and V_{total} increases with T, V_1/V_{total} eventually tends to 0. After 24 hours, the location of more than half of the users is compromised.

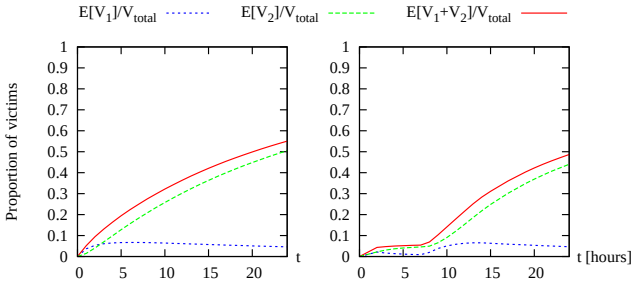

Fig. 5. Cumulative proportion of victims as a function of time. The parameters were set to: $\lambda_{\text{Arr}} = 5$ users/h, $\lambda_{\text{Dur}} = 1.5$, $\lambda_{\text{Std}} = 10$ req./h, $\lambda_{\text{LBS}} = 0.05$ req./h, and $\alpha_{\text{LBS}} = 0.2$. The dotted curve (resp. dashed) corresponds to the victims connected at (resp. arriving after) the compromise time. The solid curve represents the total proportion. Homogeneous (left) / inhomogeneous (right) Poisson processes for traffic.

Following the same line of reasoning, we can compute the number of victims with respect to the tracking attack. In this scenario, we call a victim a user who made an authenticated request during a DHCP lease for which the adversary learned – potentially *a posteriori* – the (IP,Location mapping). We assume that a user makes an authenticated request when she connects to AP (e.g., checking e-mails automatically). The average number of users who were connected at the beginning of the lease and subsequently made an authenticated request before leaving is

$$\begin{aligned}\mathbf{E}[V_1'] &= N_{\text{Con}} \int_{u=0}^{\infty} f_{\text{Dur}}(u) P_{\text{Auth}}(\min(u, T)) \\ &= N_{\text{Con}} \frac{\lambda_{\text{Auth}}}{(\lambda_{\text{Auth}} + \lambda_{\text{Dur}})} (1 - e^{-(\lambda_{\text{Auth}} + \lambda_{\text{Dur}})T})\end{aligned}$$

and the number of users who connected during the lease (and are *de facto* victims) is $\mathbf{E}[V_2'] = \lambda_{\text{Arr}} T$. To obtain the average total number of victims, we add up these two quantities and condition over the fact that the adversary obtains the mapping before the end of the lease, that is $\mathbf{E}[V'] = (1 - e^{-\Lambda_{\text{LBS}} T})(\mathbf{E}[V_1'] + \mathbf{E}[V_2'])$. The proportion of victims is depicted in Fig. 6.

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 (i.e., the probability that the adversary *links* two requests based on a common authentication token, e.g., a cookie). This occurs if at least one user makes both an authenticated request at most ΔT time

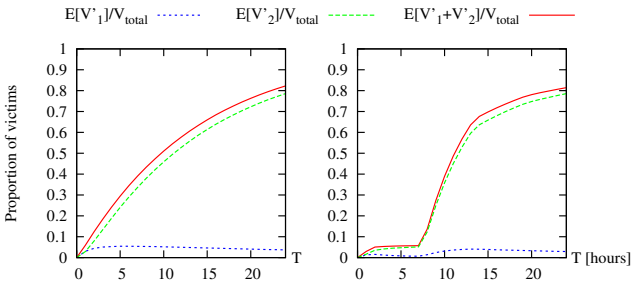

Fig. 6. Proportion of victims (wrt. the tracking attack) as a function of the duration of the lease T. The parameters were set to: $\lambda_{\text{Arr}} = 5$ users/h, $\lambda_{\text{Dur}} = 1.5$, $\lambda_{\text{Auth}} = 2$ req./h, $\lambda_{\text{LBS}} = 0.05$ req./h, and $\alpha_{\text{LBS}} = 0.2$. The dotted curve (resp. dashed) corresponds to the victims connected at (resp. arriving after) the beginning of the lease. The solid curve represents the total proportion. Homogeneous (left) / inhomogeneous (right) Poisson processes for traffic.

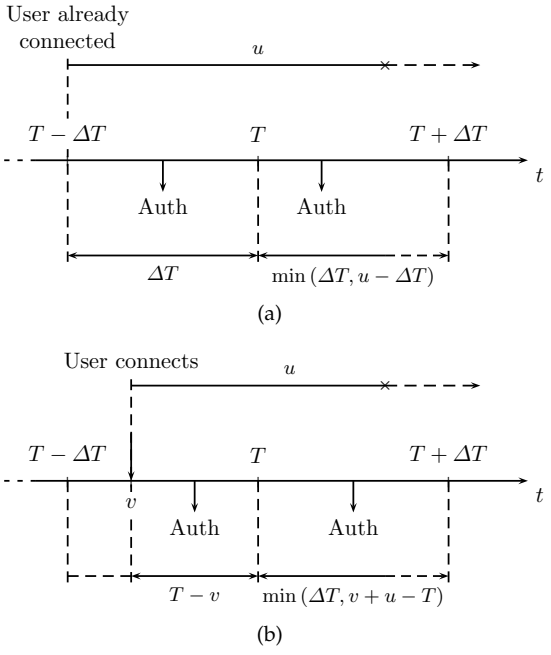

Fig. 7. Timeline for the two groups of users from which the adversary can infer the IP change. A user remains connected for a random time u. (a) The user is already connected at time $T-\Delta T$. For the adversary to infer the IP change, u needs to be greater than ΔT and the user must make at least two authenticated requests: One during the time interval $[T - \Delta T, T]$ and another one during $[T, T + \min(\Delta T, u - \Delta T)]$. (b) The user connects at some time v during the time interval $[T - \Delta T, T]$. For the adversary to infer the IP change, u needs to be greater than $T - v$ (i.e., the user must still be connected at time T) and the user must make at least two authenticated requests: one during the time interval $[v, T]$ and another one during $[T, T + \min(\Delta T, v + u - T)]$.

units ($\Delta T < T/2$) before the IP change and another authenticated request at most ΔT time units after the IP change.

Proceeding similarly as above, we compute the probability of inferring the IP change by distinguishing between two groups of users: those who were connected at time $T-\Delta T$ (see Fig. 7(a)) and those who connected within $[T-\Delta T, T]$ (see Fig. 7(b)). We denote by P_1 (resp. P_2) the probability that the adversary infers the IP change from the authenticated requests made by a user of the first group (resp. second group). First consider a user who was already connected at time $T - \Delta T$ (there are N_{Con} such users). In order to infer the IP change from the authenticated requests of such a user before time $t \in I_1$, the following conditions must be satisfied: (i) the user stays connected at least until time T, (ii) the user makes an authenticated request between the times $T-\Delta T$ and T, and (iii) the user makes an authenticated request, before time t, and before she leaves (if she leaves before time $T+\Delta T$ or until $T + \Delta T$ otherwise) (see Fig. 7). We compute the probability that at least one user (among N_{Con}) satisfies the above conditions by applying the law of total probability, conditioning over the time spent in the system from time $T - \Delta T$:

$$P_1(t) = 1 - (1 - p_1(t))^{N_{\text{Con}}}, \tag{4}$$

where

$$p_1(t) = \int_{u=\Delta T}^{\infty} f_{\text{Dur}}(u) P_{\text{Auth}}(\Delta T) P_{\text{Auth}}(\min(\Delta T, u - \Delta T, t - T))$$

Only the users that are still connected at time T can make an authenticated request in the interval $[T, T + \Delta T]$, hence u ranges from ΔT to $+\infty$.

Now consider the users who connect during the time interval $[T - \Delta T, T]$ (see Fig. 7(b)). The number of such users follows the Poisson process $N_{\text{Arr}}(\Delta T)$. By applying the law of total probability, conditioning over the number of such users, their arrival times (independent of each other and uniformly distributed within $[T-\Delta T, T]$) and their departure times, we compute the probability that at least one of the newcomers satisfies the above conditions

$$\begin{aligned} P_2(t) &= \sum_{n=1}^{\infty} \mathbf{P}\left[N_{\text{Arr}}(\Delta T) = n\right] \cdot (1 - (1 - p_2(t))^n) \\ &= 1 - e^{-\lambda_{\text{Arr}} \Delta T \cdot p_2(t)}, \end{aligned} \tag{5}$$

where

$$\begin{aligned} p_2(t) = \int_{v=T-\Delta T}^{T} \Delta T^{-1} \int_{u=T-v}^{\infty} f_{\text{Dur}}(u) P_{\text{Auth}}(T - v) \cdot \\ P_{\text{Auth}}(\min(\Delta T, v + u - T, t - T)) \end{aligned}$$

Due to space constraints, we do not include the closed-form expressions of P_1 and P_2. These can be easily computed because all integrals are of the form $\int_{u=0}^{t} u e^{-a\,u} du$, which is equal to $(1 - e^{-a\,t})/a$.

In conclusion, the probability that the adversary infers the IP change before time $t > T$, referred to

as the *linking probability*, is given by:

$$F_{\text{Link}}(t) = 1 - (1 - P_1(t))(1 - P_2(t)) \ .$$

Note that the above equations can easily be generalized to any sub-interval I_k, $k \geq 1$, by replacing $t - T$ (the relative time in I_1) by $\bar{t} = t \mod T$ (the relative time in any sub-interval).

The linking probability can be thought of as a function of both t and ΔT. Fig. 8 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. The probability P_1 (resp. P_2) of inferring the IP change from the users already connected at time $T - \Delta T$ (resp. from the users who connect after time $T - \Delta T$ and before the end of the sub-interval) first increases with ΔT: the probability of generating authenticated requests increases with the length of the interval. For large values of ΔT however (typically higher than the average connection time), P_1 decreases. This is because users connected at time $T - \Delta T$ are not likely to still be connected at time T when ΔT is large (compared to the expected connection duration $1/\lambda_{\text{Dur}}$). 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 APs in the time interval $[T - \Delta T, T + \Delta T]$ (moving from one to the other) increases with ΔT.

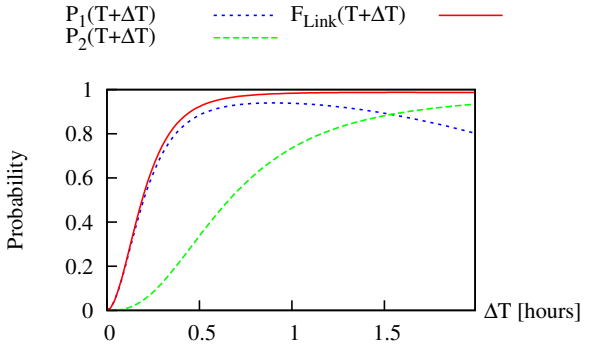

Fig. 8. Linking probability at time $T + \Delta T$ as a function of ΔT. The parameters were set to $\lambda_{\text{Arr}} = 5$ users/h, $\lambda_{\text{Dur}} = 1/1.5$, $\lambda_{\text{Auth}} = 2$ req./h, and $\alpha_{\text{LBS}} = 0.2$. The probabilities of the adversary inferring the IP change based on users connected at time $T - \Delta T$ i.e., P_1, and based on users connecting in the interval $[T - \Delta T, T]$, i.e., P_2, are represented by the dotted and dashed curves, respectively. The solid curve represents the total probability of inferring the IP change, i.e., F_{Link}.

Fig. 9 depicts the linking probability as a function of t. It remains constant for $t \geq T + \Delta T$ because only authenticated requests made before $T + \Delta T$ 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%.

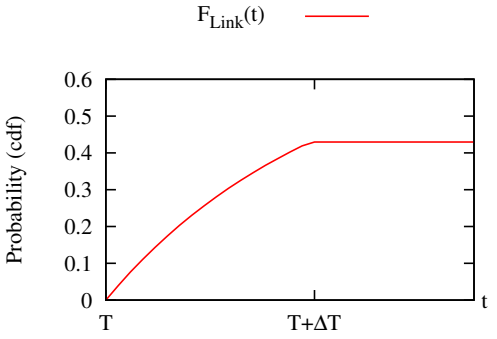

Fig. 9. Probability of inferring the IP change before time $t > T$ as a function of t, i.e., $F_{\text{Link}}(t)$. The parameters were set to: $\lambda_{\text{Arr}} = 5$ users/h, $\lambda_{\text{Dur}} = 1/1.5$, $\lambda_{\text{LBS}} = 0.05$ req./h, $\Delta T = 5$ minutes, and $\alpha_{\text{LBS}} = 0.2$.

Quantifying the threat over multiple sub-intervals. When the adversary infers the IP changes, the probability $F^{(k)}_{\text{Map}}(t)$ that the adversary knows the (IP, Location) mapping at time $t \in I_k$, $k \geq 1$ is

$$F^{(k)}_{\text{Map}}(\bar{t}) = F_{\text{Comp}}(\bar{t}) + (1 - F_{\text{Comp}}(\bar{t})) \cdot F^{(k-1)}_{\text{Map}}(T) \cdot ((1 - p_{\text{New}}) + p_{\text{New}} F_{\text{Link}}(\bar{t})) \quad (6)$$

with initial condition $F^{(0)}_{\text{Map}}(\bar{t}) = F_{\text{Comp}}(\bar{t})$: Either the adversary obtained the mapping from a LBS request during the current sub-interval (i.e., $F_{\text{Comp}}(\bar{t})$) or he did not but he obtained the mapping in the previous sub-interval and either (1) the IP did not change or (2) he inferred the IP change. Note that the assumption $\Delta T < T/2$ is required here. Indeed, this technical restriction ensures that the time interval $[kT - \Delta T, kT + \Delta T]$ (used by the adversary for the linking), does not overlap with the time interval $[(k-1)T - \Delta T, (k-1)T + \Delta T]$. Essentially, this makes the two intervals disjoint hence also independent with respect to the number of authenticated requests, which allows us to multiply the corresponding probabilities. From Equation (6), 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 equation has as a solution $a(1 - b^{k+1})/(1 - b)$. 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} in Equations (2) and (3) with the density of $F^{(k)}_{\text{Map}}$. 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. 10. It can be observed that the mapping probability increases over time and, after the convergence, the adversary successfully obtains the mapping before the lease expires in 79% of the cases and before the half-lease in 60% of the cases.

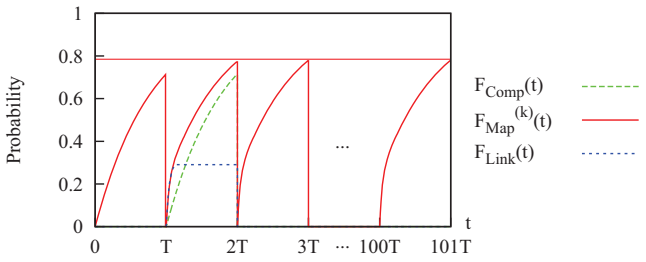

Fig. 10. Probability of obtaining the (IP, Location) mapping over several sub-intervals. The solid curve represents the probability of obtaining the mapping before time t. The dashed curve represents the probability of obtaining the mapping from an LBS request. The dotted curve represents the probability of inferring the IP change. The parameters were set to $\lambda_{\text{Arr}} = 5$ users/h, $\lambda_{\text{Dur}} = 1/1.5$, $\lambda_{\text{LBS}} = 0.035$ req./h, $\lambda_{\text{Auth}} = 0.2$ req./h, $T = 24$ h, $\Delta T = 3$ 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, this probability $F^{(k)}_{\text{Map}}(t)$ increases due to the potential inference of IP changes: it is a combination of $F_{\text{Link}}(\bar{t})$ and $F_{\text{Comp}}(\bar{t})$.

Note that in the case of the tracking attack, inferring IP change can be used to infer the location of users who connected during the past leases. Assume that the adversary could not obtain the mapping during a given lease. If he infers the IP change at the end of the lease and learns the mapping (from LBS requests) during the next lease, he can infer *a posteriori* the mapping for the previous lease and track all the users who made authenticated requests during this lease.

5 Experimental Results

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

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. To emulate the scenario of a single popular hotspot and to avoid side effects of micro-mobility, i.e., devices frequently changing the AP they are connected to, we aggregate the data of two APs located close to each other ($\sim$ 15 meters),.

Session traces contain information, obtained from RADIUS [19] logs, related to users who connect to the APs. There are three types of RADIUS events: (i) `start`: a user is successfully authenticated and the device is assigned an IP denoting the beginning of a session; (ii) `update`: a user connected to the AP periodically issues a status message; and (iii) `stop`: a user disconnects denoting the end of the session. Each entry in the log contains a timestamp, the device's anonymized MAC address, the assigned IP, the ID of the AP the device is connected to, and an event type. We observed that users typically begin arriving around 7:AM. The number of connected users peaks around 6:PM (136 on average). See diurnal patterns in the appendix. In total, 4,302 users connected over the 23 days.

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

DNS traces are obtained from the local DNS servers and each entry in the log contains a timestamp, the source IP and the requested host name. By using the source IP, timestamps and requested resources, we are able to correlate the DNS traces with the traffic traces.

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. 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 made possible by the fact that 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. The monitored services and their classification is presented in Table 1. Entries of the type *service.** refer to all the top-level-domains observed in the traces (e.g., *.com*, *.fr*, *.ca*). Entry **.gmail.com* includes *imap.*, *smtp.*, *pop.*, *www.* and *m.* and *doubleclick.** includes *.de* and *.com*. The *m.* prefix stands for mobile services.

TABLE 1
Monitored services.

Req. type	Services
Std.	www.google.*, pagead2.googlesyndication.com www.youtube.com, www.google-analytics.com, doubleclick.*, m.doubleclick.*,
LBS	maps.google.*, earth.google.com
Auth.	calendar.google.com, *.gmail.com, plus.google.com

Traffic to the monitored services (in terms of the number of 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, which shows the tremendous popularity of Google services. During a day, the average numbers of standard, authenticated and LBS requests (i.e., user-

service interactions) to the monitored services are depicted in Fig. 11. It can be observed that standard requests are prevalent, followed by authenticated requests. The moderate usage of LBS services can be explained by the location of the APs: most of the users visit this area on almost 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.

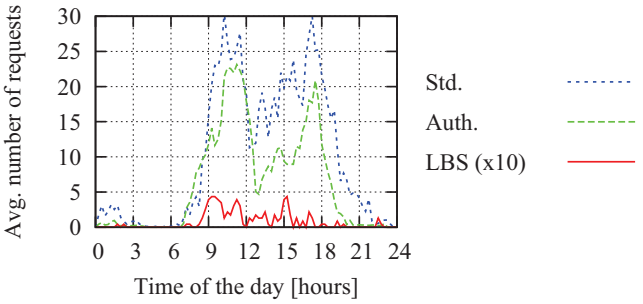

Fig. 11. Average number of standard, authenticated, and LBS requests to the monitored services over a day (averaged over 23 days).

Results. First, we measure the compromise time and the proportion of victims, by using the traces from our dataset. We compare the averaged experimental results with those from our theoretical analysis and show them in Fig. 12. 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 first connect 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 at the end of the day.

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 requests ($\alpha_{\text{LBS}} = 0.095$). Note however, that in terms of the number of users of Google services the proportion of victims actually corresponds to 90%.

Once the adversary obtains the (IP, Location) mapping, it can maintain it over time by relying on

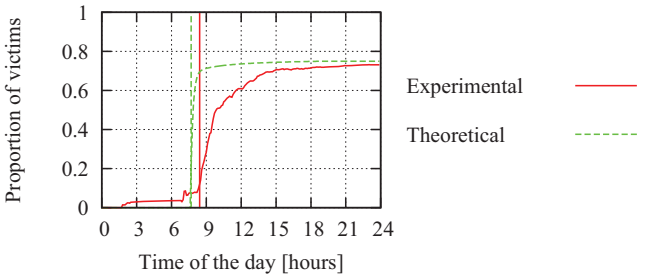

Fig. 12. Expected proportion of victims. Vertical lines represent average compromise times: theoretical T_{Comp} = 7:42:AM and experimental T_{Comp} = 8:25:AM.

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 we show the results in Fig. 13. We assume that each time the DHCP lease expires the AP is assigned a new IP address. Even with the smallest inference time window of 1 minute, the adversary can infer the IP change with the probability 1.0 between 2:PM and 5:PM. With higher values of ΔT the time during which the adversary can infer with probability 1.0 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 where there is less traffic (e.g., from 11:PM to 6:AM), the probability of the adversary inferring the mapping is smaller (< 0.2). Between 5:AM and 6:AM, the adversary cannot infer the IP change, as there is no traffic.

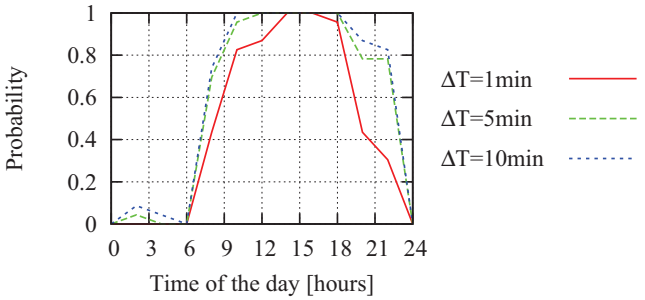

Fig. 13. Linking probability (i.e., probability of inferring the IP change) as a function of the renewal time, for different inference time window lengths (ΔT).

Consequently, the IP renewal time affects the adversary's success at maintaining the (IP, Location) mapping over time. To confirm the importance of the IP renewal time and its affect on the adversary's success, we plot the cumulative number of victims compromised at the AP during three weeks, depending on the IP renewal time (Fig. 14). We set $\Delta T = 5$ minutes

and based on the previous findings, 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. 14, we confirm that the highest number of users (3,545 out of 4,302 total number of users, which corresponds to virtually all users who access Google services) is compromised when the IP renewal happens 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). These results confirm our previous findings.

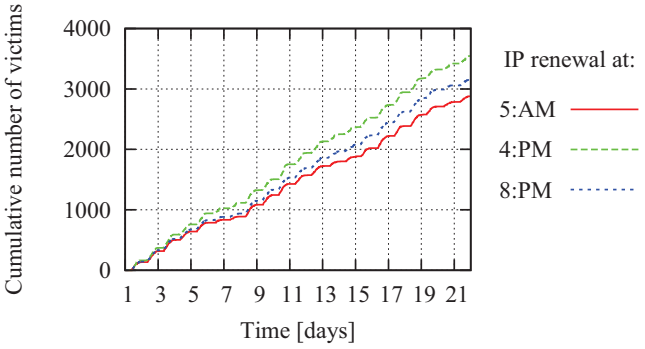

Fig. 14. Cumulative number of victims during the whole experiment, for three different IP renewal times.

6 Countermeasures

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

Hiding users' actual source IPs from the destination (i.e., the adversary) naturally comes to mind as a straightforward countermeasure against the considered threat. This 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 such a way that the destination cannot know the user's source IP. Examples of such networks include Tor [20], mix networks [21], [22], or HTTP proxies. With Virtual Private Networks (VPN), the user is assigned an IP address that belongs to a remote network (e.g., a corporate network or public/commercial VPNs). 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 [23]. Moreover, anonymization networks have a noticeable price, in terms of usability, speed and latency, that users are not willing to pay for privacy [24], [25], [26]. Note also that there are several techniques for identifying the IP of a client, even behind a NAT/proxy, e.g., by using a Java applet [27].

Alternatively, these countermeasures can be implemented by ISPs, for instance, by deploying a country-wide NAT that aggregates traffic from all hosts connected to the ISP at several gateways (e.g., Telefonica [28]) or by IP Mixing [29]. This also applies to operators of AP networks (e.g., Starbucks, AT&T Wi-Fi). However, they may not have incentives to implement such solutions.

Another approach to thwarting 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 the adversary's accuracy include spatial cloaking [30], [31] and adding noise to reported locations [32]. To increase the adversary's uncertainty, [33] proposes to inject "dummy" requests, i.e., not related to the user's location. It is not easy for users to implement these PETs, because some geolocation requests are implemented in the operating system that can be controlled by the adversary (e.g., Google Android). Moreover, when these techniques 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 could be achieved by having 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 chances 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. Beyond technical countermeasures, we envision a "Do-not-geolocalize" initiative, similar to "Do-not-track", letting users to opt-out of being localized.

7 Discussion

Scale and implications of the threat. By maintaining (IP, Location) mappings in the manner we have described, an adversary can build an IP-location system with which he can obtain (at least) sporadic user locations. For an online service provider whose goal is to profit from delivering location-targeted information, it might be sufficient to learn only current user locations at the time users access services.

However, we can envision a different type of adversary, whose goal is to mount more powerful at-

tacks on user privacy. In fact, once the adversary has access to sporadic user-location information, he is able to reconstruct entire trajectories, produce patterns of user movement habits, or infer other information about users, e.g., users' real identities, interests and activities. For example, in [34] it is shown how an adversary that observes each user's sporadic locations (that could be noisy and anonymized) can deanonymize the users, can compute the probability that a given user is at a given location at a given time, and can construct a full trajectory of each user. Golle and Partridge [35], Beresford and Stajano [36], Hoh et al. [37], and Krumm [38] use different techniques to show that users can be identified by inferring where they spend most of their time (notably their home and workplace). Our contribution is orthogonal to the aforementioned pieces of work as we *identify and study* a case of sporadic exposure whereas they study the *effect* of sporadic exposure on location privacy. In these cases, the location-privacy threat we identified serves as a building block for more powerful attacks.

Evolution of the threat with IPv6. IPv6 uses a larger address space than IPv4, hence fixing IPv4's addresses depletion issue. The wide adoption IPv6 could therefore lead to the removal of NATs. With IPv6, each host has a public IP, composed of a *prefix* (leftmost 64 bits), shared with other hosts in the same network, and a unique *host part* (rightmost 64 bits). Sharing a prefix is similar to sharing a public IPv4 address behind a NAT: (IPv4, location) mappings correspond to (prefix, location) mappings. As IPv6 prefixes are less dynamic than IPv4 addresses, the threat is amplified.

Business opportunities. Beyond threatening the location-privacy of users, the (IP, Location) mapping technique presented in this paper can be used as a novel IP-location solution that potentially improves on existing solutions [10], [39]. Online service providers, such as Google and Microsoft, are in a position to build and monetize this service by simply utilizing the 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 approach to one of the existing ones. Because ISPs control the IPs assignments 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 [40]) – some ISPs sell geographic information on the topology of their networks [27] – or by selling privacy-protection services to users.

Legal and Policy Aspects. Because the threat presented in this papers is based only on a passive analysis of the received traffic, it does not raise additional legal or policy issues, compared to what web services already do, i.e., inferring information from IPs and mining user traffic to improve the offered services.

8 Conclusion

In this paper, we have presented a practical threat, effectively demonstrating that the location privacy of users connected to access points can be (unintentionally) compromised by others. The scale of the threat is significant because it simply relies on the way most networks are designed (i.e., using NAT). When successful, the service provider locates 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 effect on the adversary's success. This framework serves as a lightweight alternative to an extensive traffic analysis for estimating the threat. We experimentally investigate the state in practice, by analyzing real traces (collected from deployed Wi-Fi access points) of users of Google services. 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 users' location privacy. However, 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, in particular the adversary's inference of the AP's precise location based on *all* the LBS requests it receives (in this paper, we assumed that the adversary learns the mapping as soon as he receives one LBS request), (ii) the adversary's inference of IP changes, taking into account e.g., fingerprinting users, and the trade-off between the probability of inferring the IP change and the adversary's confidence, and (iii) the evolution of the threat for mobile users, in particular the adversary's ability to track users as they move and connect to different APs..

Acknowledgements

The authors are very grateful to Yves Despond and Patrick Timsit, from the IT department, for their valuable help in collecting data from the Wi-Fi network of EPFL, and to Alevtina Dubovitskaya for her effort.

References

[1] N. Vratonjic, K. Huguenin, V. Bindschaedler, and J.-P. Hubaux, "How Others Compromise your Location Privacy: The Case of Shared Public IPs at Hotspots," in *PETS*, 2013.

[2] S. Patil, G. Norcie, A. Kapadia, and A. Lee, ""Check Out Where I Am!": Location-Sharing Motivations, Preferences, and Practices," in *CHI*, 2012.

[3] G. Goodell and P. Syverson, "The right place at the right time," *Communications of the ACM*, vol. 50, no. 5, pp. 113–117, 2007.

[4] PricewaterhouseCoopers, "Internet Advertising Revenue Report," 2011.

[5] "Targeting Local Markets: An IAB Interactive Advertising Guide," Interactive Advertising Bureau, 2010.

[6] E. Katz-Bassett, J. P. John, A. Krishnamurthy, D. Wetherall, T. Anderson, and Y. Chawathe, "Towards IP Geolocation Using Delay and Topology Measurements," in *IMC*, 2006.

[7] Z. Qian, Z. Wang, Q. Xu, Z. M. Mao, M. Zhang, and Y.-M. Wang, "You Can Run, but You Can't Hide: Exposing Network Location for Targeted DoS Attacks in Cellular Networks," in *NDSS*, 2012.

[8] "Geolocation and Online Fraud Prevention from MaxMind," http://www.maxmind.com/.

[9] "HostIP: My IP Address Lookup and Geotargeting Community Geotarget IP Project," http://www.hostip.info/.

[10] I. Poese, S. Uhlig, M. A. Kaafar, B. Donnet, and B. Gueye, "IP Geolocation Databases: Unreliable?" *ACM SIGCOMM Computer Communication Review*, vol. 41, pp. 53–56, 2011.

[11] Google Engineering Center Zurich, "Technology and Innovation for Web Search," Private communication, Oct. 2012.

[12] M. Casado and M. J. Freedman, "Peering Through the Shroud: The Effect of Edge Opacity on IP-Based Client Identification," in *NSDI*, 2007.

[13] USA Department of Defenses, "Global Positioning System: Standard Positioning Service Performance Standard," 2008.

[14] "Skyhook Location Perf." http://www.skyhookwireless.com/location-technology/performance.php.

[15] "Google Privacy Policy," http://www.google.com/intl/en/policies/privacy/preview/, 2012.

[16] M. J. Freedman, M. Vutukuru, N. Feamster, and H. Balakrishnan, "Geographic Locality of IP Prefixes," in *IMC*, 2005.

[17] S. M. Ross, *Stochastic Processes*. Wiley, 1995.

[18] A. Ghosh, R. Jana, V. Ramaswami, J. Rowland, and N. Shankaranarayanan, "Modeling and Characterization of Large-Scale Wi-Fi Traffic in Public Hot-Spots," in *INFOCOM*, 2011.

[19] C. Rigney, S. Willens, A. Rubens, and W. Simpson, "Remote Authentication Dial In User Service," RFC 2865, IETF, 2000.

[20] R. Dingledine, N. Mathewson, and P. Syverson, "Tor: The Second-generation Onion Router," in *USENIX Security*, 2004.

[21] D. L. Chaum, "Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms," *Communications of the ACM*, vol. 24, no. 2, pp. 84–90, 1981.

[22] G. Danezis, R. Dingledine, D. Hopwood, and N. Mathewson, "Mixminion: Design of a Type III Anonymous Remailer Protocol," in *S&P*, 2003, pp. 2–15.

[23] "Tor Metrics Portal," https://metrics.torproject.org.

[24] A. Acquisti, R. Dingledine, and P. Syverson, "On the economics of anonymity," in *FC*, 2003, pp. 84–102.

[25] R. Dingledine and N. Mathewson, "Anonymity loves company: Usability and the network effect," in *WEIS*, 2006.

[26] B. Fabian, F. Goertz, S. Kunz, S. Mller, and M. Nitzsche, "Privately waiting – a usability analysis of the tor anonymity network," in *Sustainable eBusiness Management*, 2010, pp. 63–75.

[27] J. A. Muir and P. C. V. Oorschot, "Internet Geolocation: Evasion and Counterevasion," *ACM Computing Survey*, vol. 42, pp. 4:1–4:23, 2009.

[28] "Telefonica Implements NAT for ADSL Users," http://bandaancha.eu/articulo/7844/usuarios-adsl-movistar\-compartiran-misma-ip-mediante-nat-escasear-ipv4, 2012.

[29] B. Raghavan, T. Kohno, A. C. Snoeren, and D. Wetherall, "Enlisting ISPs to Improve Online Privacy: IP Address Mixing by Default," in *PETs*, 2009.

[30] M. Gruteser and D. Grunwald, "Anonymous Usage of Location-Based Services Through Spatial and Temporal Cloaking," in *MobiSys*, 2003.

[31] C. A. Ardagna, M. Cremonini, S. De Capitani di Vimercati, and P. Samarati, "An Obfuscation-Based Approach for Protecting Location Privacy," *IEEE TDSC*, vol. 8, no. 1, pp. 13–27, 2011.

[32] R. Agrawal and R. Srikant, "Privacy-Preserving Data Mining," in *SIGMOD*, 2000.

[33] H. Kido, Y. Yanagisawa, and T. Satoh, "An Anonymous Communication Technique using Dummies for Location-Based Services," in *ICPS*, 2005, pp. 88–97.

[34] R. Shokri, G. Theodorakopoulos, J.-Y. Le Boudec, and J.-P. Hubaux, "Quantifying Location Privacy," in *S&P*, 2011.

[35] P. Golle and K. Partridge, "On the Anonymity of Home/Work Location Pairs," in *Pervasive*, 2009.

[36] A. Beresford and F. Stajano, "Location Privacy in Pervasive Computing," *IEEE Perv. Comp.*, vol. 2, pp. 46–55, 2003.

[37] B. Hoh, M. Gruteser, H. Xiong, and A. Alrabady, "Enhancing Security and Privacy in Traffic-Monitoring Systems," *IEEE Perv. Comp.*, vol. 5, pp. 38–46, 2006.

[38] J. Krumm, "Inference Attacks on Location Tracks," in *Pervasive*, 2007.

[39] Y. Wang, D. Burgener, M. Flores, A. Kuzmanovic, and C. Huang, "Towards Street-Level Client-Independent IP Geolocation," in *NSDI*, 2011.

[40] CNN, "Your Phone Company is Selling Your Personal Data," http://money.cnn.com/2011/11/01/technology/verizon_att_sprint_tmobile_privacy, 2011.

Nevena Vratonjic is a cyber-security program manager at Kudelski Security, Lausanne, Switzerland. Her expertise is in network security, game theory, online advertisement and privacy. She earned a M.Sc. in computer and communication sciences from University of Belgrade in 2006 and a Ph.D. from EPFL in 2013.

Kévin Huguenin is a post-doctoral researcher at EPFL. His research interests include performance, security and privacy in networks and distributed systems. He earned a M.Sc. degree from Ecole Normale Supérieure de Cachan and the Université de Nice – Sophia Antipolis, France, in 2007 and a Ph.D. in computer science from the Université de Rennes, France, in 2010.

Vincent Bindschaedler is a Ph.D. student in computer science at the University of Illinois at Urbana-Champaign (UIUC). His research focuses on security and privacy in distributed systems, and cloud security. He earned a B.Sc. and a M.Sc. in computer science, from EPFL in 2010 and 2012, respectively.

Jean-Pierre Hubaux is a professor at EPFL (which he joined in 1990). His current research activity is focused on privacy, notably in pervasive communication systems. In 2008, he completed a graduate textbook, entitled Security and Cooperation in Wireless Networks, with Levente Buttyan. He held visiting positions at the IBM T.J. Watson Research Center and at UC Berkeley. He is a fellow of both the ACM and IEEE.

APPENDIX A
NOTATIONS

TABLE 2
Table of notations.

Symbol	Definition
T	DHCP lease time
p_{New}	Probability of being assigned a new IP
I_k	k-th sub-interval
$\bar{t}$	Relative time within a sub-interval
λ_{Arr}	Rate of user arrivals at AP
$N_{\mathrm{Arr}}(t)$	Number of arrivals in an interval of length t
T_{Dur}	Time users stay connected to the AP
$1/\lambda_{\mathrm{Dur}}$	Avg. time users stay connected to the AP
$F_{\mathrm{Dur}}, f_{\mathrm{Dur}}$	Pdf/cdf of T_{Dur}
N_{Con}	Avg. number of users connected to the AP
$\lambda_{\mathrm{Std}}, \lambda_{\mathrm{Auth}}$	Rates of user std./auth. requests
$P_{\mathrm{Std}}(t)$,$P_{\mathrm{Auth}}(t)$	Probability that a user makes at least one request during an interval of length t
α_{LBS}	Proportion of users who make LBS requests
λ_{LBS}	Rate of user LBS requests
Λ_{LBS}	Aggregated rate of users' LBS requests
T_{Comp}	First time an LBS request occurs
$F_{\mathrm{Comp}}, f_{\mathrm{Comp}}$	Pdf/cdf of T_{Comp}
W	Length of the vulnerability window
ΔT	Time interval used to infer IP changes
$F_{\mathrm{Link}}(t)$	Probability of knowing IP change at time t
$F^{(k)}_{\mathrm{Map}}(t)$	Probability of having the mapping before time $t \in I_k$

APPENDIX B
DATASET STATISTICS

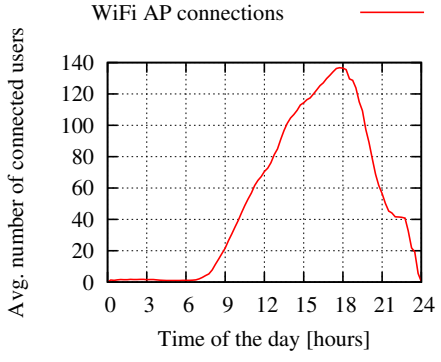

Fig. 15. Average number of users connected to the AP over a day (averaged over the 23 days).