SecureRun: Cheat-Proof and Private Summaries for Location-Based Activities

Anh Pham, *Student Member, IEEE,* Kévin Huguenin, *Member, IEEE,* Igor Bilogrevic, *Member, IEEE,* Italo Dacosta, *Member, IEEE,* Jean-Pierre Hubaux, *Fellow, IEEE*

Abstract—Activity-tracking applications, where people record and upload information about their location-based activities (*e.g.,* the routes of their activities), are increasingly popular. Such applications enable users to share information and compete with their friends on activity-based social networks but also, in some cases, to obtain discounts on their health insurance premiums by proving they conduct regular fitness activities. However, they raise privacy and security issues: the service providers know the exact locations of their users; the users can report fake location information, for example, to unduly brag about their performance. In this paper, we present SecureRun, a secure privacy-preserving system for reporting location-based activity summaries (*e.g.,* the total distance covered and the elevation gain). SecureRun is based on a combination of cryptographic techniques and geometric algorithms, and it relies on existing Wi-Fi access-point networks deployed in urban areas. We evaluate SecureRun by using real data-sets from the FON hotspot community networks and from the Garmin Connect activity-based social network, and we show that it can achieve tight (up to a median accuracy of more than 80%) verifiable lower-bounds of the distance covered and of the elevation gain, while protecting the location privacy of the users with respect to both the social network operator and the access point network operator(s). The results of our online survey, targeted at RunKeeper users recruited through the Amazon Mechanical Turk platform, highlight the lack of awareness and significant concerns of the participants about the privacy and security issues of activity-tracking applications. They also show a good level of satisfaction regarding SecureRun and its performance.

Index Terms—Activity tracking; Location privacy; Social networks; Location proofs

1 Introduction

More and more people rely on activity-tracking applications to monitor, manage and to encourage themselves to do physical activities. Mobile apps, such as Endomondo, Garmin Connect, RunKeeper, Runtastic and Strava, and wearable devices, such as Fitbit, Nike+ Fuelband and Jawbone UP, enable users to keep track of their performance while running, hiking or cycling. This information is collected using location-based services (LBSs) and embedded sensors in smartphones and wearable devices. Due to the popularity of these apps, top mobile operating systems now include APIs that facilitate the gathering and sharing of fitness and health data across multiple apps and devices (*e.g.,* HealthKit for iOS and Google Fit for Android). A key feature of these applications is to enable users to access summaries of their activities and performance statistics and to share this information with other users and service providers on online social networks. For instance, users can share the total distance covered, the cumulative elevation gain and the path taken during their activities. For this purpose, activity-tracking applications collect and send users' location and fitness data, possibly while they pursue their activities, to services providers.

In exchange for their data, users are offered various incentives. For example, users can receive discounts, coupons or even cash [2]–[6], awards at competitions [7], [8] or simply points to improve their social reputation. In addition, many companies, including big names such as British Petroleum (BP), Bank of America and Autodesk, are giving activity-tracking devices to their employees to encourage healthier lifestyles and, as a result, improve productivity and lower corporate insurance costs [9]–[11]. Similarly, health insurance companies such as United Health, Kaiser Foundation Group, Humana Group and Aetna have created programs to include activity-tracking devices into their policies, *i.e.,* consumers are rewarded by the insurers with lower rates based on their activity summaries [6], [10], [12].

Although activity-tracking and sharing services are gaining popularity, there are two important issues that could hinder their wide-scale adoption and viability. First, the privacy concerns associated with the data collected by these apps. In particular, users' location data, which is known to service providers, can be used to infer private information about them, such as their home/work locations [13], [14], activity preferences [15], interests [16] and social networks [17]. This risk is exacerbated by the fact that service providers often share or sell this data to third-parties [18], [19]. Second, the concerns about the integrity of the data reported by users. As the value of incentives increase, users

- *Manuscript received: September 18, 2015;*
- *This article is a revised and extended version of a paper that appears in the Proceedings of the 16th ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp 2014) Pham* et al. *[1].*
- *Anh Pham, Italo Dacosta, and J.-P. Hubaux are with EPFL, Lausanne, Switzerland (e-mail: thivananh.pham@epfl.ch, italo.dacosta@epfl.ch; jean-pierre.hubaux@epfl.ch).*
- *Kévin Huguenin is with LAAS-CNRS, Toulouse, France (e-mail: kevin.huguenin@laas.fr). This work was partially carried out while the author was with EPFL, Lausanne, Switzerland.*
- *Igor Bilogrevic is with Google Inc., Zurich, Switzerland (e-mail: ibilogrevic@google.com). This work was carried out while the author was with EPFL, Lausanne, Switzerland.*

could be more tempted to cheat when reporting their performance [20], which would endanger the viability of the system for the service provider and its affiliates, as well as its attractiveness to the users. For example, location cheating can be achieved by making mobile devices report erroneous location information to the activity-tracking app [21], [22], or by spoofing the GPS/Wi-Fi signals used for geo-location users [23]–[25]. Moreover, some tools enable users to manipulate activity data to lie about their performance [26]–[28].

To assess the awareness and concerns of users of activity-tracking applications regarding opportunities to cheat and privacy issues, we conducted a user survey of 50 participants. Our survey participants are active RunKeeper users who we recruited on the Amazon Mechanical Turk platform. In the survey questionnaire, we first informed the participants about existing opportunities to cheat and privacy issues of fitness-tracking applications such as RunKeeper, and we then polled them about their awareness and their concerns (see Section 6.1 and Appendix F of the supplemental material for more details, including the full transcript of the questionnaire). Regarding opportunities to cheat, we found that all the participants were unaware of them and 48% were very or extremely concerned about them. Regarding privacy issues, we found that 90% of the participants were unaware of them and 82% were very or extremely concerned about them. These results highlight the need to raise awareness and the need for technical solutions to build cheat-proof and private activity-tracking apps.

A straightforward solution to these issues consists in enforcing the use of either secure and/or privacy-preserving location proofs for users [23], [29]–[31], where their location could be either (1) trusted and known (as it is the case for activity trackers) or (2) untrusted and known (but useless for obtaining rewards). In fact, solutions guaranteeing (1) would benefit the service provider by ensuring that cheating is infeasible, whereas solutions satisfying (2) would protect users' location privacy but would provide locations that are too coarse-grained for computing meaningful summaries.

In this paper, we propose SecureRun, a novel infrastructure-based approach that provides guarantees both in terms of the prevention of cheating and location privacy for the users *vis-à-vis* the service provider, while allowing the latter to compute accurate summaries and statistics of users' activities, such as the total distance covered during an activity. SecureRun relies on existing wireless access-point (AP) networks and alleviates the need for a costly deployment of a dedicated ad-hoc infrastructure. Instead, it could rely on strategic partnerships between social network providers and access-point network operators. SecureRun consists of two phases: First, users obtain secure and privacy-preserving proofs of performance during their activities, by relying on a lightweight message exchange protocol between their mobile device and the Wi-Fi access points encountered while pursuing the activity; second, the service provider computes an accurate summary of a user's activity, such as the total distance covered between two time instants or the elevation gain, without learning any additional information about the user's actual location. SecureRun produces, in a privacy-preserving way, a secure and accurate lower bound of the actual distance covered by a user while she performs an activity. Finally, it is able to take advantage of the co-existence of multiple access-point operators to improve the accuracy/privacy trade-off. Unlike the initial version of our system presented in [1], in order to maximize the accuracy of the activity summaries produced, SecureRun computes the optimal set of access points to communicate with. To the best of our knowledge, this is the first work to address privacy and cheating issues in the computation of activity summaries.

We evaluate our solution on a large data-set of real user-activities, collected from the Garmin connect [32] social network in the regions of Brussels (Belgium), London (UK) and Paris (France). For these regions, we also extract the actual locations of a network of deployed Wi-Fi APs operated by FON [33]. Moreover, to evaluate the benefits of having multiple operators in a given area, we extract the locations of a second network in the urban area of Paris. The experimental results show that our solution can gracefully balance accuracy and privacy. SecureRun achieves good accuracy, up to a median accuracy of more than 80%. This constitutes a significant improvement compared to the performance of the initial version of the system reported in [1]: On average, we achieved an absolute improvement of 5-10 points in the median accuracy. In our survey, we also asked the participants about their level of satisfaction regarding activity summaries for different values of the summary accuracy (i.e., "Assuming that you have run 10 miles, what would your level of satisfaction be if SecureRun issues a certificate of 6 mi? 7 mi? 8 mi? 9 mi?): We found that for an accuracy of 80% (i.e., SecureRun issues a certificate of 8 mi when the user runs for 10 mi), which roughly corresponds to the median performance of SecureRun, the "high" and "very high" levels of satisfaction account for 42% of the participants (74% with the "medium" level of satisfaction). We also conduct a sensitivity analysis to evaluate the effect of the distribution of the APs on the performance of SecureRun.

The remainder of the paper is organized as follows. We first survey the related work and introduce the system and adversarial models. We then present SecureRun and report on its evaluation in terms of its performance and of its security and privacy properties. Finally, we present directions for future work and conclude the paper.

2 Related Work

Cheating on activity-based social networks is becoming a serious problem. For example, He et al. [23] show that users can easily override Foursquare's GPS verification mechanisms by modifying the values returned by the calls to the geo-location API of smartphones. Similarly, Polakis et al. [34] use a black-box approach to uncover the mechanisms used by Foursquare and Facebook Places to detect location attacks and propose several ways to circumvent them. Moreover, work from Carbunar and Potharaju [20] analyze data from Foursquare and Gowalla and find that incentives to cheat exist because people actively check-in and collect rewards. Thus, it is necessary to carefully balance incentives with a more effective verification of users' location claims. In this regard, Zhang et al. [24] show that fake check-ins lead not only to monetary losses for the venues offering special deals on location-based check-ins but also to the degradation of the quality of service provided by recommendation systems that rely on users' location information. Carbunar et al. [31] also show that there is tension between privacy and correctness in location-based applications, where users are unable to prove that they have satisfied badge conditions without revealing the time and location of their check-ins.

Meanwhile, to defend against cheating, researchers have also proposed several mechanisms that offer secure verification of location information. From a broad perspective, such mechanisms can be grouped in three categories: infrastructure-independent, infrastructure-dependent and hybrid mechanisms. In the *infrastructure-independent* approach, a user obtains location evidence from her neighbors by using short-range communication technologies, such as Bluetooth [35]–[37]. Specifically, Talasila et al. [35] propose a location authentication protocol where a set of users help verify each others' location claims. The protocol operates by keeping a centralized authority that, based on users spatio-temporal correlation, decides whether such claims are authentic or not. Similarly, Zhu et al. [36] propose a system where mutually co-located users rely on Bluetooth communications to generate their location claims that are then sent to a centralized location verifier. In addition to the security and privacy guarantees presented in [35], Zhu et al. [36] enable individual users to evaluate their own location privacy and decide whether to accept location proof requests by other users. Jadliwala et al. [38] provide a formal analysis of the conditions needed in an ad-hoc network to enable any distance-based localization protocols in wireless networks. Similar approaches have been explored in mobile sensor networks [39], [40].

More in line with our work, the *infrastructure-dependent* studies assume the presence of a centrally-operated set of access points (AP) that produce and verify location claims. For instance, to ensure the presence of a user in a given region, the AP can require her to execute together a nonce-based, challenge-response protocol, with constraints on the maximum round-trip delay of the messages exchanged between the user and the AP [41], or any distance bounding protocol [42]–[45], which enables the AP to check the minimum distance between itself and the user. In particular, Capkun and Hubaux [44] propose a verifiable multilateration protocol that can be used to securely position nodes in a wireless network. Once the secure localization phase is done, the user can obtain a location proof to certify that the user is at a specific geographical location [41]. Alternatively, Luo and Hengartner [30] show that a user can obtain location proofs with different precision levels and then select one to disclose to the service provider, depending on her privacy preferences.

Hybrid approaches rely on both landmarks (*e.g.,* WiFi, cellular base stations) and short-range communications between users (*e.g.,* Bluetooth) to obtain location evidences. For instance, Uchiyama et al. [46] describe an opportunistic localization algorithm for positioning mobile users in urban areas. The area of presence of a user is calculated based on a map of obstacles (*i.e.,* the area where there is no radio signal), the last presence areas of the node itself and the nodes it encountered. Similarly, Koo et al. [47] present a hybrid system that relies on landmarks located at corners or intersections of streets and short-range communication between users. The system considers the users' routes (*i.e.,* a sequence of segments connecting successive landmarks), the average moving speed in the segment and the collaboration between mobile nodes to locate the users.

SecureRun relies on an infrastructure of wireless access points to provide secure distance proofs, in line with the infrastructure-dependent models discussed above. However, it is the first work, to the best of our knowledge, to provide secure distance proofs and to tackle the challenge of activity summaries.

3 System Architecture

In this section, we describe the different entities involved in our system: a user, one or more Wi-Fi network operator and a service provider (e.g., a social network). Figure 1 depicts the system considered and a sketch of SecureRun. We also describe the adversarial model in this scenario. Moreover, we discuss and analyze the incentives of the various involved entities and the adoption of SecureRun in Section 6. For the sake of readability, we provide a table with the notations used throughout the paper in Appendix A.

3.1 Users

We assume that some users pursue location-based activities, where they move in a given geographical region, and that they want to obtain statistics or summaries of their activities. These users are equipped with GPS- and WiFi-enabled devices and have sporadic Internet connectivity (at least at some point in time before and after the activity). Thus, they can locate themselves and communicate with nearby Wi-Fi access-points. We assume a unit-disc model for Wi-Fi communications, in which a user and an AP can communicate only if the distance between them is lower than a given radius R, which is constant across all users and all APs. Note that we do not assume that users can communicate with the APs as soon as the distance is lower than R (we only assume that *if* they can communicate with an AP, *then* the distance between the user's device and the AP is lower than R); as such, this is a relaxed version of the unit-disc model. Note also that this assumption can always be enforced by choosing a high value of R. In particular, we assume that users cannot violate this model by, for example, increasing the transmission power of their devices. We assume that users can obtain random identifiers (or pseudonyms) from the online service provider, and that they can use such pseudonyms to protect their privacy while pursuing their activities. We assume that users do not hand their pseudonyms to other users (this can be enforced by embedding sensitive or critical information about the users in their pseudonyms, such as tokens that enable the users to reset their passwords). Finally, we assume direct Wi-Fi connections to have much smaller communication delays than cellular Internet connections, thus enabling us to prevent proxy/relay attacks [23] by using delay-based challenge-response mechanisms.

In order to brag or obtain rewards, users might be tempted to unduly increase their performance by deviating from the protocol. To do so, such users could, for instance, report locations that are different from their actual locations, forge messages or reuse messages they, or their friends, obtained in the past.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2015.2483498, IEEE Transactions on Mobile Computing

3.2 Wi-Fi AP Network Operator

We assume the existence of one or multiple Wi-Fi network operators, and that each operator controls a set of fixed Wi-Fi APs deployed in the regions where the users pursue their activities. ISP-controlled Wi-Fi community networks such as British Telecom Wi-Fi (or a federation of such networks, such as FON) constitute typical candidates for deploying and running the APs used by SecureRun, because they own and control the routers they provide to their subscribers (e.g., the ISPs can transparently remotely update the firmware of their subscribers' routers). Each AP is aware of its geographic position and of its communication radius. We assume that all the APs have synchronized clocks, and that they are able to compute public-key cryptographic operations. In particular, we assume that all the APs from a same network operator share a public/private group key pair $(GK_{\text{pub}}, GK_{\text{priv}})$, where GK_{pub} is known by the users and the service provider, and GK_{priv} is only known to the network operator and to its APs. Finally, we assume that the APs cannot uniquely identify mobile devices based on the characteristics of their wireless-network adapters (*i.e.,* wireless fingerprinting techniques based on clock skews or physical layer signatures [48]–[50]).

Some access-point operators might be interested in tracking the users' locations, based on the information obtained by all of their APs.[1] We assume them to be *semi-honest* or *honest-but-curious*, meaning that they do not deviate from the protocol specified in our solution, but that they analyze the information they collect while executing the protocol.

3.3 Social Network Provider

We assume that there is a social-network provider that offers activity summaries and sharing services to its registered users. The provider is able to generate sets of pseudonyms for its users, by using a suitable public-key encryption scheme. Moreover, it is able to verify the authenticity of messages signed with the network operators' group keys (by using their public group keys). Like the network operators, the social network provider might be interested in the users' locations[1] and is assumed to be *honest-but-curious*.

Finally, we assume that the different entities do not collude with each other.

4 SecureRun

Our high-level design goal is to build an activity-tracking system that (1) guarantees the authenticity of the user's activity data with respect to cheating users who try to unduly increase their performance and (2) protects the users' location-privacy with respect to curious network operators and service providers that try to track them.

In this section, we present SecureRun, our solution for secure and privacy-preserving activity summaries. First, we give a high-level overview of SecureRun and define the main operations it involves. Then, we provide a detailed description of each of the aforementioned operations. Figure 1 shows an overview of SecureRun and of its operations.

4.1 Overview

From a general perspective, SecureRun operates as follows. As a user pursues her location-based activity, she moves and communicates (through her smartphone) with the wireless access points located along her route (and in her communication range) to obtain *location proofs* (LP). A location proof is a digitally signed message, delivered by an access point, that certifies that the user is, at a given time t, in a given range of an access point that is located at a given position (x, y).[2] The times/positions at which users request such location proofs are determined by a *sampling algorithm*.

A user employs different pseudonyms (provided to her beforehand by the service provider) when communicating with the access points. The different location proofs obtained by a user (from different access points) in a short interval of time are *aligned in time* and *combined* into a more precise location proof by using intersection techniques. To obtain an *activity proof*, a user provides pairs of consecutive precise location proofs to an access point; more specifically, she obtains a *distance proof* (DP) and/or an *elevation proof* (EP). The activity proofs that the user obtains do not provide any location information, as they do not include information about where the activity was pursued but only about the distance or elevation. Such proofs are digitally signed messages that certify that a user achieved (at least) one given performance during a given time span, *e.g.,* that she ran at least 1 km between 3:02pm and 3:08pm on March 19th. Finally, a user sends all the activity proofs she collects, while pursuing her activity, to the service provider that performs the adequate verifications; if the information is correct, the provider combines the proofs into an activity summary that it publishes on the user's profile.

In terms of privacy, the use of pseudonyms protects users' locations with respect to the access-point operators; the lack of location information in activity proofs provides protection with respect to the social-network provider. Finally, the use of digital signatures and pseudonyms, combined with the fact that the activity proofs represent the lower bounds of the user's actual performance, provide security properties with respect to dishonest users.

4.2 Location Proofs

At each sampling time t_i (determined by the sampling algorithm described below), a user begins to collect location proofs from the access points in her communication range. To do so, she periodically broadcasts (during a short time interval starting at time t_i) location-proof requests that contain one of her pseudonyms P. Note that a different pseudonym is used for each sampling time. All the access points in her communication range send back messages that contain the pseudonym P, a timestamp t (*i.e.,* the time at which the request is processed by the access point) and their coordinates (x, y), digitally signed with the private group key GK_{priv}, specifically a location proof $\text{LP} = \text{sig}_{GK_{\text{priv}}}\{P, t, (x, y)\}$. We denote by $\text{LP}_{i,j} = \{P_i, t_{i,j}, (x_{i,j}, y_{i,j})\}$ the j-th location proof collected at sampling time t_i (note that we omit the

[1] The information available to the adversaries is made explicit in the description of the protocol (Section 4) and summarized in the security analysis (Section 7).

[2] Throughout the paper, we use an equi-rectangular projection to map the latitude and longitude of the considered locations to a Cartesian coordinate system, in which the Euclidean distance is a good approximation of the Haversine distance.

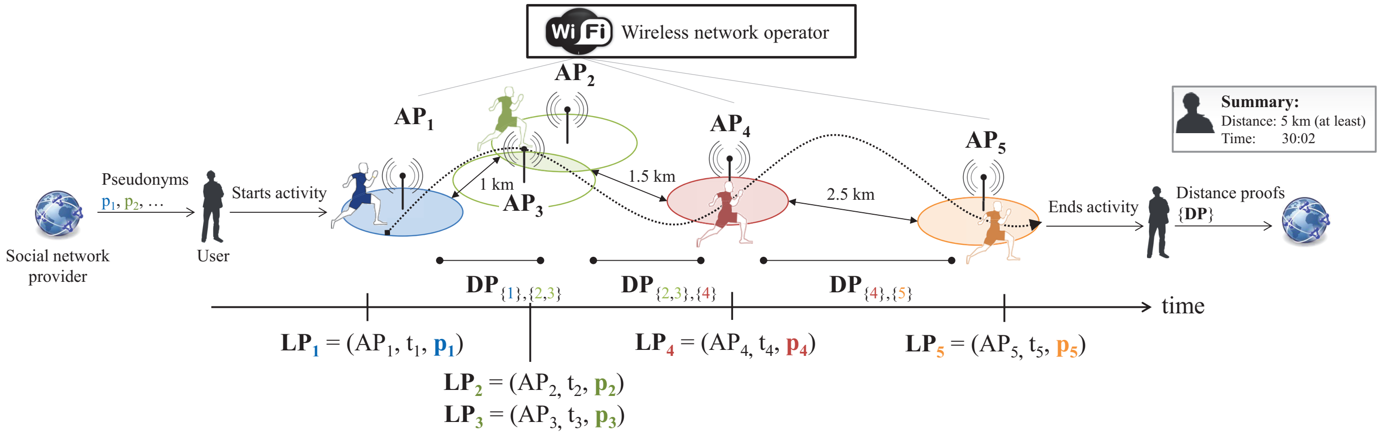


Figure 1. Overview of SecureRun's system architecture. A user first obtains a set of pseudonyms $\{p_1, \ldots, p_K\}$ from the social network provider. Then, while performing an activity along the dotted trajectory, she sporadically requests location proofs (LP) at times τ_i, using pseudonyms p_i, to the APs encountered along the trajectory. By using the LPs, the APs compute, and deliver to the user, distance proofs for the different time intervals. The user finally sends the distance proofs to the social network provider that combines them and publishes the summary on her profile.

signature for the sake of readability). As the communication and processing delays differ from one access point to another, the location proofs collected from different access points at a same sampling time have different timestamps. Under the relaxed unit-disc communication model (with radius R), such a location proof certifies that, at time t, the user is at a distance of at most R to the access point that issues the location proof. In other words, it certifies that the user is in a disc of radius R, centered at the point of coordinate (x, y). We denote such a disc by $\mathcal{C}((x, y), R)$. In addition, we denote by C the entire set of APs located along the user's route and by C_i the set of APs defined by the location proofs collected by the user at time t_i.

4.3 Activity Proofs

To obtain an activity proof (*i.e.*, a distance proof or an elevation proof), a user sends to any access point (whenever she needs it) the location proofs collected at two consecutive sampling times t_i and t_{i+1}. The contacted AP first combines the location proofs (those collected at each of the two sampling times) into more precise location proofs, by aligning them in time and intersecting them. As these location proofs have different timestamps, the first step of the combination consists in aligning the different location proofs as follows. Assuming the speed at which users move is upper-bounded by a constant $v_{\max}$, the fact that a user is at a distance at most d to an access point at time t means that, at time t', the user is at a distance of at most $d + v_{\max} \cdot |t - t'|$ to this access point. The second step of the combination simply consists in computing the intersection of the aligned location proofs. Note that only the locations proofs with a timestamp in $[t_i, t_i + \delta t]$ are combined. The access point determines a geographical area A_i where the user was at time t_i from the following expression[3]

$$A_i = \bigcap_{j \in \mathcal{C}_i} \mathcal{C}((x_{i,j}, y_{i,j}), R + v_{\max} \cdot |t_i - t_{i,j}|) \tag{1}$$

The AP repeats the same operation for the location proofs obtained at sample time $i + 1$. Figure 2 shows an example of a location-proof combination: Assuming that at sampling time t_1, a user obtains two LPs: $\text{LP}_{11} = \{P_1, t_{11}, (x_{11}, y_{11})\}$ and $\text{LP}_{12} = \{P_1, t_{12}, (x_{12}, y_{12})\}$ from two APs.

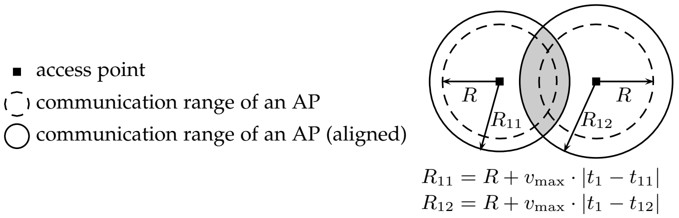

■ access point
◌ communication range of an AP
○ communication range of an AP (aligned)
R
R
R11
R12
$R_{11} = R + v_{\max} \cdot |t_1 - t_{11}|$
$R_{12} = R + v_{\max} \cdot |t_1 - t_{12}|$

Figure 2. Time alignment of location proofs LP_{11} and LP_{12} at time t_1. At t_1, the user was in the gray area defined by the intersection of the two solid discs.

The activity proofs are computed from a lower bound of a user's performance. As for distance proofs, knowing that a user was in an area A_i at time t_i and in an area A_{i+1} at time t_{i+1}, the distance d_i between A_i and A_{i+1} (*i.e.*, the minimum of the distances between any point in A_i and any point in A_{i+1}) constitutes a lower bound of the distance covered by a user during the time interval $[t_i, t_{i+1}]$. More specifically, using the Euclidean distance, we have

$$d_i = \min_{\substack{(x,y) \in A_i \\ (x',y') \in A_{i+1}}} \sqrt{(x - x')^2 + (y - y')^2} \tag{2}$$

A tight approximation of d_i can be obtained by using a nonlinear optimization toolbox such as IPOPT [51].

With respect to the elevation proofs, the following expression gives a lower bound of the cumulative elevation gain[4] achieved by a user during the time interval $[t_i, t_{i+1}]$.

$$e_i = \min_{\substack{(x,y) \in A_i \\ (x',y') \in A_{i+1}}} (\max(0, z(x', y') - z(x, y))) \tag{3}$$

[3] Note that we do *not* consider the information that the user is not in the communication ranges of the APs from which she did not receive location proofs. This case is discussed in Appendix E.

[4] Note that the elevation loss can be computed by following the same line of reasoning.

where $z(\cdot,\cdot)$ denotes the elevation of the point of coordinate (x, y). Note that the "max" operator is used here in order to account for only *positive* elevation gains. Unlike for the lower bound of the covered distance, we compute the lower bound of the elevation gain analytically: $e_i = \max\left(0, \min_{(x,y)\in A_{i+1}} z(x,y) - \max_{(x,y)\in A_i} z(x,y)\right)$. Figure 3 illustrates the stages of the generation of activity proofs in the case of the covered distance/elevation gain.

To provide stronger privacy-guarantees, SecureRun also hides the time intervals $[t_i, t_{i+1}]$ included in each activity proof, by encrypting them with an order-preserving encryption scheme (OPE) [52], [53]. As its name implies, an OPE scheme guarantees that the numerical ordering of the input data is preserved in the encrypted output data, *i.e.*, $a > b \Leftrightarrow E_{\mathrm{OPE}}(a,k) > E_{\mathrm{OPE}}(b,k)$, where $E_{\mathrm{OPE}}()$ is an OPE encryption function and k is the corresponding encryption key. In a nutshell, without the key, the encrypted timestamps cannot be decrypted but can still be ordered.

SecureRun uses OPE as follows. When an AP generates an activity proof, it encrypts the timestamps with the key GK_{OPE} (known to all APs) and includes them in the proof, instead of the plain-text timestamps. Thus, the access point generates an activity proof $\mathrm{sig}_{GK_{\mathrm{priv}}}\{d_i, e_i, [c_i, c_{i+1}], \{P_i, P_{i+1}\}\}$ where $c_i = E_{\mathrm{OPE}}(t_i, GK_{\mathrm{OPE}})$ and $c_{i+1} = E_{\mathrm{OPE}}(t_{i+1}, GK_{\mathrm{OPE}})$.

4.4 Activity Summary

To publish an activity summary on her profile, the user uploads her collected activity proofs to the social network service provider. In turn, the provider checks that (1) the signatures of the activity proofs are valid (using the public group keys of the access points), that (2) all the pseudonyms that appear in the activity proofs indeed belong to the user, and that (3) the OPE-encrypted time intervals of the activity proofs do not overlap (otherwise the distance covered in the time overlap would be counted twice, hence violating the lower-bound property of the summary). If this is the case, the social network provider simply sums the distances (or the elevation gains, respectively) from the activity proofs and adds the resulting summary to the user's profile. Notice that, thanks to OPE, the service provider can validate that the time intervals are indeed disjoint, without learning the exact time values.

If the user's mobile device has an active wireless Internet connection (*e.g.*, 3/4G), the user can upload on-the-fly the activity proofs she collects while she pursues the activity. By doing so, it enables real-time online tracking of the user's activity by her friends, such as the LIVE-tracking feature offered by the Runtastic and Endomondo mobile apps.

4.5 Sampling Algorithms

We now describe SecureRun's sampling algorithm. The sampling algorithm determines the sampling times/positions at which the user requests location proofs from the access points in her communication range. The general objective of the sampling algorithm is to achieve high accuracy (*i.e.*, tight lower-bounds in the activity proofs) and a high level of privacy.

We distinguish between two cases: the case where a user knows beforehand the path she is about to take, namely *planned sampling*, and the case where she does not, namely *unplanned sampling*. In both cases, the sampling algorithm knows the locations of the access points. Planned sampling corresponds to the quite common situation where a user records the set of her preferred paths and of her past activities. Such a feature is commonly implemented in activity-tracking applications (including Garmin's) in order to enable users to *compete* against their own previous performance. For instance, the activity-tracking application indicates to the user whether she is late or in advance, compared to her best performance. With planned sampling, the sampling points are determined before the user starts the activity with the full knowledge of the path, thus yielding potentially better results. We now describe both variants of the algorithm, considering at first the case of a single access-point operator and, subsequently, multiple operators.

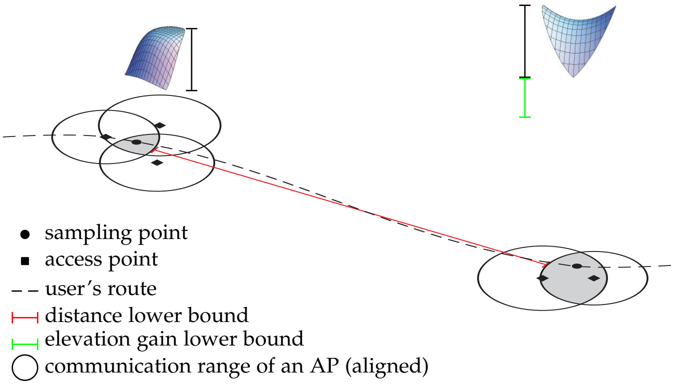


Figure 3. Computation of distance and elevation proofs. The shaded areas correspond to the intersections of the location proofs obtained at the same sampling time. The 3D plots correspond to the elevation profiles of the shaded areas, based on which the lower-bound of the elevation gains are computed.

We focus our description on the case of distance proofs.[5] The planned and unplanned versions of the algorithm share a common design rationale: (1) limit the discrepancies between the actual path and the lower-bounds, by requesting location proofs where the direction of the path changes significantly; and (2) enforce a silence period after requesting certain location proofs, in order to achieve unlinkability of successive activity proofs.

Silence Periods. To highlight the importance of silence periods, consider a user who collects three location proofs at three successive sampling times (with pseudonyms P_1, P_2 and P_3). If she requests a distance proof for the time interval between the first two location proofs and another distance proof for the time interval between the last two, the access-point operator can link the three location proofs (as it knows that P_1 and P_2 belong to the same user and so do P_2 and P_3) and thus track the user, despite the use of pseudonyms. To circumvent this issue, a user requests an additional location proof some time after she requests the second location proof, leaving her with four location proofs. The time between the second and the third (*i.e.*, the additional) location proofs is called a *silence period*. Finally, the user requests distance proofs only for the time intervals between the first and the second, and between the

[5] The reasoning is the same for elevation proofs.

third and the forth location proofs. The distance covered between the second and the third location proofs is not counted in the user's activity summary. The users repeat this process throughout her activity, as depicted in Figure 4. The duration ΔT of the silence period[6] is a parameter of the system that enables users to balance their accuracy of the activity summaries and their privacy: Short silence periods yield high-accuracy activity summaries (as the distances covered during the silence periods, which are not counted in the activity summary, are small) but provide low privacy guaranties (as the access-point operators can link with high confidence two successive activity proofs, because the time interval between them is short). Conversely, long silence periods yield low-accuracy activity summaries and provide high privacy guaranties.

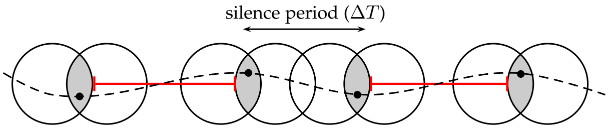


Figure 4. Silence period. By implementing a silence period between every pair of successive distance proofs (*i.e.*, not requesting a distance proof for this period), a user reduces the risk of her distance proofs being linked by the access-point, hence protecting her privacy.

Multiple Access-Point Operators. In the case where multiple access-point operators are involved, the silence periods are not always needed: By requesting successive distance proofs from different operators (assumed to not collude with each other), a user does not need to wait for ΔT seconds (*i.e.*, implement a silence period) to reduce the risks of linking her distance proofs. To protect her privacy, a user should not need to request two successive distance proofs from the same operator, unless she implements a silence period. With two operators, a user can alternatively request distance proofs from each of the two operators, as illustrated in Figure 5.

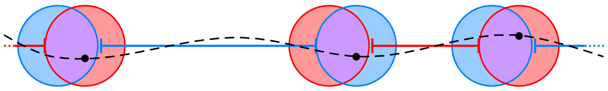


Figure 5. Case of multiple access-point operators (Operator 1 in red and Operator 2 in blue). At every sampling point, a user requests location proofs from both operators. Then, she requests distance proofs, alternatively from different operators, to reduce the risk of linking the distance proofs she collects without reducing the accuracy of her activity summary (unlike with silence periods).

For the sake of simplicity, we now describe the planned and unplanned sampling algorithms without silence periods, in the case of a single access-point operator.

Planned sampling. In the planned case, we formalize the sampling problem at discrete time: The path of a user is represented as a sequence of time/position samples, ordered by increasing times $\{(\tau_i, p_i)\}_{i=1..T}$. Such a discrete formalism matches the conventional format of activity traces (*e.g.*, Garmin represents the paths of activities as sequences of time/positions samples). In such a formalism, the sampling problem consists in selecting sampling times (*i.e.*, , a subset of $\mathcal{T} = \{\tau_i\}_{i=1..T}$) at which a user needs to collect location proofs from the neighboring access points so that the sum of the resulting activity proofs is maximized. Using the notations from Equations 1 and 2, the sum of the distance proofs obtained by collecting locations proofs at times $\tau_{i_1}, \ldots, \tau_{i_K} \in \mathcal{T}$, with a single operator and without silence periods, is

$$\sum_{k=1}^{K-1} d\left(A_{i_k}, A_{i_{k+1}}\right), \tag{4}$$

and the optimal set of sampling points is the subset of $\mathcal{T}$ that maximizes this quantity.[7] This optimization problem is equivalent to a maximum-weight path problem in a directed graph G, where the vertices of G are the time samples (*i.e.*, $\mathcal{T}$), and where the edges of G connect each time sample τ_i to all the subsequent time samples τ_j ($j > i$), with a weight equal to the value of the distance proof obtained by combining the location proofs obtained at time τ_i, with those obtained at time τ_j (*i.e.*, $d(A_i, A_j)$). A sample graph is shown in Figure 6, together with an example of a set of sampling points (in red). As G is a directed acyclic graph, the maximum-weight path problem can be solved in linear time in the number of edges, that is $\mathcal{O}(T^2)$. Unlike the heuristic proposed in [1], this formalization enables SecureRun to find the optimal set of sampling points, resulting in an absolute improvement of 5-10 points in the median accuracy of SecureRun (see Section 5).

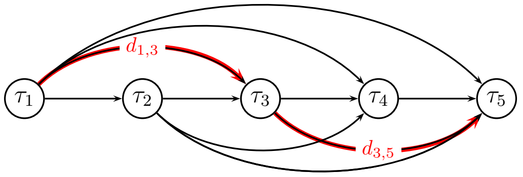


Figure 6. Graph construction for the maximum-weight path formulation of the sampling problem (one operator, no silence periods). The thick, red path shows an example of a set of sampling points: The user collects location proofs at times τ_1, τ_3, and τ_5. The total of the obtained distance proofs is $d_{1,3} + d_{3,5}$ ($d_{i,j}$ is the short for $d(A_i, A_j)$).

The sampling point problem can be formulated as a maximum-weight graph problem also with silence periods and with multiple operators. The details of the graph construction in these different scenarios are given in Appendix B of the supplemental material.

Note that in practice, a user normally does not follow the exact same path as she previously did. Therefore, the algorithm determines sampling points based on the previously recorded path, and the user requests location proofs when she reaches the *vicinity* of a pre-determined sampling point (*e.g.*, within 20 m).

Unplanned Sampling. In the unplanned version, only the current and past positions of the user are known to the algorithm. A users first collects location proofs at the starting point of her activity (*e.g.*, when she presses the "start" button

[6] In practice, the length of the silence period is a random variable of mean ΔT (*e.g.*, drawn for the uniform distribution on $[0.5\Delta T, 1.5\Delta T]$) in order to prevent an access-point operator from linking two distance proofs based on the time elapsed between them.

[7] The exact same reasoning applies to elevation-gain proofs. The distance function $d()$ that appears in Equation 4 should be replaced with the elevation-gain function $e()$ from Equation 3.

on her mobile device). As the user pursues her activity, the algorithm periodically determines whether location proofs should be requested. To do so, the algorithm compares the actual distance covered since the last sampling point with the straight-line distance between the last sampling point and the current position. If the difference between the two distances is higher than a threshold, the algorithm triggers the collection of location proofs. To limit the rate at which location proofs are collected, we impose a minimal distance between two sampling points. Algorithm 1 presents the pseudo-code version of the unplanned sampling algorithm, where $d(\cdot\,,\,\cdot)$ denotes the distance between two locations.

Algorithm 1 Unplanned sampling algorithm.

```
Input: MIN_LP                      ▷ Minimum distance between two LPs
       MAX_LP                      ▷ Maximum distance between two LPs
       MAX_ERR                                         ▷ Maximum error
$S \leftarrow [\,]$                ▷ List of past locations since last sampling
while true do
    $p_c \leftarrow$ current location
    $S \leftarrow S + [p_c]$
    $p_l \leftarrow S[1]$

    if $d(p_l, p_c) <$ MIN_LP then
        next

    $e \leftarrow \left(\sum_{k=1}^{|S|} d(S[k], S[k+1])\right) - d(p_l, p_c)$

    if $d(p_l, p_c) >$ MAX_LP or $e >$ MAX_ERR then
        sample()
        $S \leftarrow [p_c]$
```

4.6 Summary

In this section, we have presented SecureRun, a solution for providing secure and privacy-preserving activity summaries, and we have described in detail the different operations it involves. The inaccuracy of the activity summaries, defined as the difference between the lower bounds and the actual values, produced by SecureRun are due to the fact that (1) the distances covered inside the areas $\{A_i\}$ and the distances covered during the silence periods are not counted, and (2) the paths taken by the users between two areas are approximated with a straight line. We will report on the evaluation of the accuracy of SecureRun in the next section. The security and privacy properties of SecureRun are provided by the use of pseudonyms and cryptographic techniques, by the aggregation and sanitization of data (with respect to location information), and by the silence periods. We discuss this in more detail in Section 7.

5 Evaluation

We evaluate SecureRun's performance on real traces of users' activities from Garmin Connect [32], pursued in cities where wireless access-point networks are deployed by the FON operator [33] (and possibly Free [54]). We consider scenarios where mobile users, equipped with Wi-Fi-enabled devices, report the total elevation gain and distance covered during their location-based activities (e.g., running). We focus our evaluation on three geographical areas that correspond to the cities of Brussels, London and Paris. We also discuss the practical aspects of the implementation of SecureRun on wireless routers.

5.1 Data Sets

In order to evaluate SecureRun, we collected data sets of access-point locations and activities, and we relied on the Google Elevation API. All the data we collected was publicly available and the data collection was in accordance with the policies of the ethical committee at EPFL (HREC).

Wi-Fi Access Points. In late 2013, we collected the geographic coordinates of the Wi-Fi access points from the FON community network in the region of Brussels, London and Paris. FON is a large community network with more than 14 million hotspots (12 million at the time of the data collection) worldwide, most of them are located in western Europe. FON achieves very high coverage in urban areas (up to 2,500 AP/km^2) through strategic partnerships with local ISPs (*e.g.,* Belgacom, British Telecom, SFR): The routers of the ISPs' subscribers, provided by the partner ISP, act as FON hotspots. As ISPs hold total control over the routers of their subscribers (through automatic firmware updates), they could easily implement and deploy SecureRun. Overall, we obtained the locations of 92,280 unique APs in Brussels, 39,776 unique APs in London, and 87,521 unique APs in Paris. In order to evaluate SecureRun with multiple access-point network operators (used jointly as described in the previous section), we also collected the geographic coordinates of the Wi-Fi access points from the Free community network. Free is a major French national ISP that offers community network features based on the routers of its subscribers. We obtained the locations of 60,280 unique APs from Free in Paris, which correspond to a density of 445±381 AP/km^2. Figure 14 (top) and Figure 12 (shown in Appendix C) depict the heat-maps of the densities of FON access points and Free access points respectively . We can observe that the density of access points is low in regions corresponding to rivers, cemeteries, parks, highways and railways; this is due to the community nature of the FON network (*i.e.,* APs are in residential areas).

Activities. In early 2014, we collected activity information from Garmin Connect, an online service where users can upload and share information about their location-based activities, including the type of activity (*e.g.,* running, biking) and the path of the activity (under the form of time-coordinates samples). We collected *running* activities and we computed, for each of them, the duration, the length and the cumulative elevation gain of the path, the inter-sample times, and the density of APs along the path (*i.e.,* the number of APs met along the path, assuming a unit-disc communication model with a radius $R = 25$ meters, normalized by the length of the path). For each activity, we divided its path into chunks of 500 m, and we determined for each chunk whether it is covered by at least one access point (i.e., it intersects with the communication range of at least one access point). This metric is crucial for SecureRun to work, as having a high proportion of covered chunks ensures that users will be able to collect location proofs, and thus distance proofs. To exclude clear outliers or activities that are not covered by a minimal number of access points from our data set, we filtered out activities that (1) last less than 10 minutes or more than 4 hours, or (2) are shorter than 2 km or longer than 45 km, or (3) have a gap of more than 10 minutes between two samples, or (4) have less than 4 AP/km along their paths, or (6) have less than 20% of covered chunks. In the remainder of the paper, we consider only the activities that pass the aforementioned filters (*i.e.,* the *filtered data-sets*). Table 1 summarizes the filters applied to our raw data-set.

Table 1
Summary of the filters applied to our activity data-set.

Property	Filter
Duration	<10 min or > 4 h
Length	<2 km or > 45 km
Elevation gain (only for elevation gain summaries)	> 50 m
Inter-sample times	> 10 min
Density of AP along activities	< 4 AP/km
Proportion of covered chunks	< 20%

Figure 15 (shown in Appendix C of the supplemental material) shows the experimental cumulative distribution functions of the main characteristics of the activities used in our evaluation and Figure 14 (bottom) (shown in Appendix C) depicts the heat-maps of the densities of activities (*i.e.,* the number of distinct activities that cross a given area of the map). It can be observed that many activities take place in parks, where the density of access points is relatively low. In the filtered data set, we observe a median inter-sample time of 3-4 seconds (which correspond to 7-11 meters). Table 4 (shown in Appendix C) summarizes some relevant (w.r.t. our solution) statistics on the filtered data.

Elevation. We evaluate SecureRun for the case of elevation only in the region of Paris, and we filter out the activities with bogus elevation data or with an elevation gain lower than 50 m. In order to determine the minimum and maximum elevation in a given region, typically the intersection of communication discs centered at the AP locations, we rely on the Google Elevation API and sample the elevation of random points in the regions of interest. We make sure that every region contains at least 20 samples. We obtained a total of 701,793 such elevation samples. We show the elevation map of Paris in Figure 13 included in Appendix C.

5.2 Methodology

We implement SecureRun in a Java simulator and evaluate its performance for the activities from the Garmin Connect data-set, with the access-point networks from the FON and Free data-sets (under the relaxed unit-disc communication model with a radius of 25 meters). The parameters used in our simulation are shown in Table 2. For each activity, we simulate the execution of SecureRun in different scenarios: with one or multiple access-point network operators, with the planned/unplanned sampling algorithm, and for different values of the parameters (such as the duration ΔT of the silence periods). For each such setting, we compute the corresponding activity summary. We measure the SecureRun's performance in terms of the *accuracy* of an activity summary: the ratio between the distance (resp. elevation) in the summary and the actual distance (resp. elevation) covered by the user during her activity.

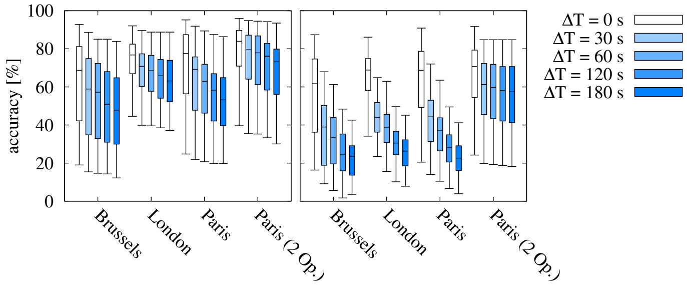
Table 2
Parameters used in the simulation.

Parameter	Description	Value	Unit
$v_{\max}$	Maximum speed	3	m/s
R	Communication range of APs	25	m
MAX_LP	Maximum distance between two LPs	500	m
MIN_LP	Minimum distance between two LPs	50	m
MAX_ERR	Maximum relative error	10	m
δt	Threshold on the time difference between aligned LPs	5	s

5.3 Results for Distance Summaries

First, we look at SecureRun's absolute performance in different settings. Figure 7 shows a box-plot representation (first quartile, median, third quartile, and outliers) of SecureRun's accuracy in the (a) planned and (b) unplanned cases, in the cities of Brussels, London and Paris, for different durations of the silence periods. In the case of Paris, we also evaluate SecureRun with two access-point operators (Free and FON).

Overall, SecureRun achieves good performance: up to a median accuracy of 78% (Paris, FON, planned sampling, $\Delta T = 0$). This value drops to 69% when unplanned sampling is used. It can be observed that, as expected, the planned sampling algorithm yields consistently better results than the unplanned algorithm, and that the accuracy decreases with the duration of the silence period. In the case of two operators (in Paris), it can be observed that the accuracy is substantially better (84%) compared to the scenario with a single operator, when the duration of the silence period is set to 0. This is because a user can optimize the lengths of her distance proofs between the two operators. Moreover, the (negative) effect of the duration of the silence periods on the accuracy is substantially lower in the case of two operators (73% for the case of two operators vs. 53% for the case of a single operator, with planned sampling and ΔT=180 s). This is because silence periods are less frequently needed in such a scenario, only when a user requests a distance proof from an operator and cannot find any access points belonging to the other operator for the subsequent distance proof). Finally, the performances are quite similar across the different cities, with a slight advantage for London, which has a higher proportion of covered chunks. This confirms our intuition and suggests that SecureRun's performance increases with the proportion of covered chunks. Compared to the initial version of SecureRun, we observed an absolute improvement of 5-10 points in the median accuracy. The improvement is lower for situations where the accuracy is already good, typically when no silence periods are used (except in the case of two operators). This means that the optimal sampling algorithm is most beneficial for adjusting the silence periods (except in the case of two operators in which they are less needed) and for choosing a best operator when there are several of them. Note that, as the number of Wi-Fi access points and their communication ranges are likely to increase in the coming years, SecureRun's performance is expected to increase. Moreover, with the emergence of activity-tracking apps, the deployment of networks of wireless access points (including in parks) dedicated to activity tracking could become a thriving business, thus further increase the performance and applicability of SecureRun.



(a) planned sampling (b) unplanned sampling

Figure 7. Accuracy of the distance summaries, with the (a) planned and (b) unplanned sampling algorithms, for different values of the duration of the silence periods, with the FON network (+ Free for Paris 2 operators).

To further study the sensitivity of SecureRun to the density and the distribution of the access points (as captured by the number of AP/km and the proportion of covered chunks, respectively), we split the activities into three buckets, based on the values of these two metrics, and we plot the experimental cumulative density functions of the accuracy in each of these buckets. Activities with a low density of AP and/or a low proportion of covered chunks typically correspond to those that are located in parks; thus they do not really match our target context, *i.e.,* urban areas. In the case of two operators, we only consider the values of the metrics with respect to the FON network.

The results are depicted in Figure 8, with planned sampling and $\Delta T = 60$ s. It can be observed that the performance is substantially better for high densities and for high proportions of covered chunks, as compared to the low counterparts. In Brussels for instance, the median accuracy goes up to 74% for activities with high densities, whereas it is only 57% for all the activities. Note that even for some activities with a high density, the accuracy can be quite low (*i.e.,* <25%). We investigated this issue by manually inspecting the paths of these activities and we report on the results in Appendix D of the supplemental material.

5.4 Results for Elevation-Gain Summaries

Figure 9 shows a box-plot representation of the accuracy of our solution for the case of elevation gain in Paris, with the planned sampling algorithm, for different durations of the silence periods. Overall, SecureRun achieves reasonable performance: up to a median accuracy of 60 % with one operator and silence periods of 60 seconds. This number goes up to 78 % when two operators are used.

5.5 Practical Aspects of SecureRun

For SecureRun to have low latency, it should be implemented at a low layer, typically at the MAC layer. Moreover, the computations involved in the AP-part of the protocol (e.g., cryptography, optimization) should be lightweight.

To better understand the technical challenges of the implementation of SecureRun on a router and to assess its feasibility, we developed a proof-of-concept prototype on a router (alix2d2 @ 500 MHz, 256 MB of RAM) running OpenWrt by using the Click framework[8] for network operations and OpenSSL for cryptographic operations. To do so, we defined a new frame at the MAC layer of the Wi-Fi protocol. In our experiments, we observed a delay of 9 milliseconds for the two-way message exchange between a user's mobile device and an AP to obtain an LP (for an ECDSA signature algorithm with a 224-bit key). The router managed to handle 12 clients that simultaneously send one LP request per second. As for the message exchange to obtain DPs, we estimated the additional running time incurred by the computation of a DP with IPOPT at 150 milliseconds (i.e., we ran it on a laptop and made a projection for a router based on the CPU speed). Note that as IPOPT's optimization algorithm is iterative, we can easily balance the running time and the accuracy of the computation. Note also that the computation of the DPs could be advantageously offloaded to a cloud server maintained by the AP network operator. Finally, to put the aforementioned delays in perspective, note that, in our dataset, a user remains in the connection range of an AP for about 10.7 seconds on average (i.e., the total delay for a message exchange between the user and the AP for a LP/DP is less than 2% of the in-range time).

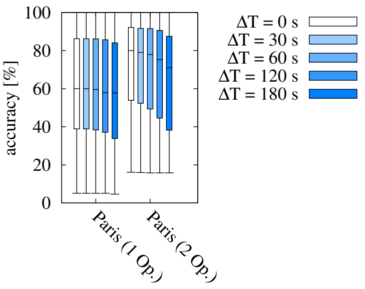


Figure 9. Accuracy of the elevation gains proofs in Paris, with the planned sampling algorithms, for different values of the duration of the silence periods, with the FON network (+ Free for 2 operators).

As for the battery consumption overhead induced by SecureRun on the mobile device, SecureRun essentially consists of two main operations: the sampling algorithm and LP/DP collection. The unplanned sampling algorithm is lightweight and the planned sampling algorithm can be run offline (and only once per route). The LP/DP collection process is sporadic and involves cryptographic operations, the overhead of which is negligible compared to those performed by activity-tracking applications (e.g., continuous geolocation, HTTPS communication over 3G/4G).

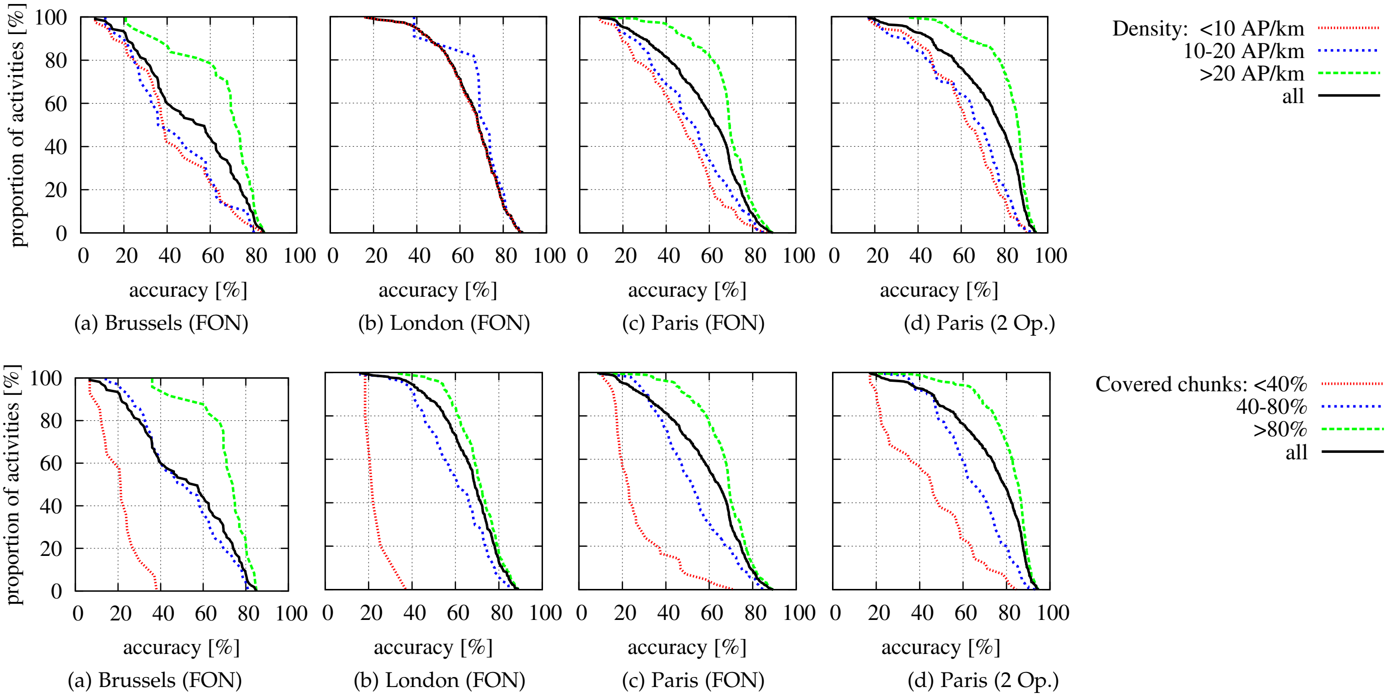
6 Adoption of SecureRun

In order for SecureRun to be adopted, it must be beneficial to all the stakeholders: the users, the Wi-Fi AP network operator, and the social-network provider. In this section, we discuss and analyze the incentives for the stakeholders.

6.1 Users

In order to assess the interest, as well as the expectations, of activity-tracking application users in SecureRun, we conducted a user survey in early 2015. Our survey involved 50 participants recruited through the Amazon Mechanical Turk platform. The online survey was composed of 11 questions covering general demographics, awareness and

[8] http://www.read.cs.ucla.edu/click



(a) Brussels (FON) (b) London (FON) (c) Paris (FON) (d) Paris (2 Op.)

(a) Brussels (FON) (b) London (FON) (c) Paris (FON) (d) Paris (2 Op.)

Figure 8. Sensitivity analysis of the accuracy, with respect to the density of access point along the activities (top) and to the proportion of covered chunks (bottom). The plots represent complementary cumulative distribution functions (ccdf). The planned sampling algorithm was used, with silence periods of $\Delta T = 60$ s. Note that in London, all activities have a density $\leq$20 AP/km.

concerns about opportunities to cheat and privacy issues in activity-tracking systems, and satisfaction and expectations regarding the accuracy of cheat-proof and private summaries (see Appendix F of the supplemental material for the full transcript of the survey questionnaire). The survey took approximately two minutes to complete; we paid each participant US $2. In order to be allowed to participate in our survey, participants were required to have an active RunKeeper account[9] and a minimum Human Intelligence Task (HIT) approval rate of 95% with at least 100 past approved HITs. We obtained a diverse and balanced sample of participants: 44% of male and 56% of female, diverse primary areas of employments, age ranging from 20 to 47 year old (avg.: 31, stdev.: 5.9). Most participants used a single activity-tracking application.

We polled the participants about their awareness of applications that enable users to cheat (i.e., claim performances they did not actually achieve) and about their concerns regarding the authenticity of the performance reported by their friends. None of the participants were aware of the existence of such applications; however, a large proportion of the participants (48%) were very or extremely concerned about them. Similarly, we polled the participants about their awareness of the privacy implications of activity-tracking applications (i.e., the ability to infer home/work locations, demographic information, and medical conditions, and the fact that activity data can be sold to third parties such as insurance companies[10]) and about their concerns regarding these issues. Only 10% of the participants were aware of the existence of such privacy issues; a large majority of the participants (82%) were very or extremely concerned about it. These results clearly make the case for SecureRun.

Finally, we briefly introduced SecureRun to the participants and we polled them about their satisfaction (on a 5-point Likert scale) if, for a 10-mile run, SecureRun provided them with a certificate of 5, 6, 7, 8 or 9 miles. We show the results in Figure 10. It can be observed that for an accuracy of 80% or more (which is the case for SecureRun in most scenarios), the participants' opinions were mostly positive.

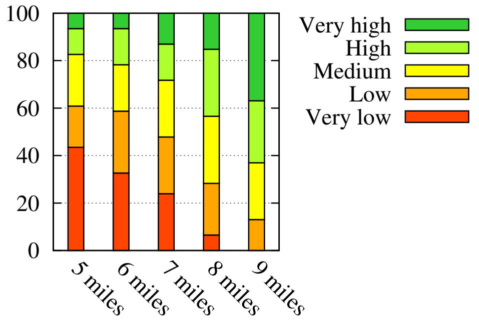


Figure 10. Level of satisfaction of the survey participants for various values of the accuracy of activity summaries.

6.2 Wi-Fi AP-Network Operators and Service Providers

Social network providers[11] offering activity-tracking applications have incentives to deploy SecureRun as it increases

[9] An active account is an account with at least 20 running activities, recorded with the mobile apps, with a duration of at least 10 minutes and a distance of at least 2 miles over the time period spanning from 2013-01-01 to 2014-12-01. In order for us to check this, the participants had to grant us access to their RunKeeper account through the API

[10] http://www.dailymail.co.uk/news/article-2409486/

[11] The same applies to health insurance companies that provide their customers with activity-tracking apps.

their attractiveness to the users and certifies the data provided by their users. In order to deploy SecureRun, the social-network provider needs the help of one or multiple Wi-Fi AP-network operators. The social-network provider could pay the AP-network operator for the certified activities, in a B2B fashion, either directly (i.e., in currency) or through advertisement. For instance, we envision a business model in which the activities that are certified by distance proofs provided by the FON network and uploaded on Garmin Connect or RunKeeper are displayed with an icon "Certified by FON", thus providing FON with ad space on the activity-based social network website. The AP-network operator could also charge the distance proofs to the users (e.g., via a specific subscription). Finally, with the emergence of activity-tracking apps and inexpensive low-power Bluetooth beacons (e.g., iBeacons), wireless networks dedicated to activity-tracking could be deployed (e.g., in parks).

7 Security and Privacy Analysis

In this section, we discuss the security and privacy guarantees provided by SecureRun, by considering the adversarial model from Section 3 (*i.e.,* three possible adversaries).

7.1 Cheat-Proof Guarantees

Adversary: User. Malicious users can attempt to cheat by forging or modifying location or activity proofs. To prevent these types of attacks, SecureRun relies on digital signatures to protect the integrity of the proofs created by the APs. Moreover, users cannot double-count some of the distances they cover, because the service provider validates that the time intervals on each activity proof do not overlap before summing them up. Note that as the service provider does not allow users to declare activities that overlap in time, a user cannot use twice a collected distance proof. Users could also try to claim other users' location or activity proofs as their own. However, the service provider can detect such cheating attempts by checking that the pseudonyms included in the activity proofs belong to the user claiming them. A malicious user could also ask other users to collect proofs on her behalf by sharing her pseudonyms. To discourage such attempts, pseudonyms include sensitive information such as tokens to reset the user's password or modify certain service's settings.[12] Also note that pseudonyms are sent encrypted to the APs with their group public key to prevent eavesdropping by other parties and that our threat model assumes honest-but-curious APs (*i.e.,* they do not abuse the information on the pseudonyms).

SecureRun also requires users to be in the communication range with the APs to obtain valid location proofs. Users can try to bypass this restriction by launching proxy attacks, in which two users collude to obtain valid location proofs. Still, such attacks can be limited by introducing constraints on the execution time of the protocol (*i.e.,* distance-bounding mechanisms). For instance, the AP operator could impose a communication delay on the Wi-Fi interface that is smaller than the one achieved on cellular networks.

Finally, the activity proofs obtained are, by design, lower-bounds of the performance achieved by the users. Thus, independently of the way users obtain and combine their location proofs, the reported summary will always be lower than the actual performance.

7.2 Privacy Guarantees

Adversary: Service Provider. A curious service provider could try to infer location information from its users' activity proofs. However, activity proofs contain neither location information nor time information (*i.e.,* timestamps are obfuscated by using order-preserving encryption[13]), hence the service provider cannot link the activity proofs to actual locations or time intervals. In a region covered by APs, a given distance (more precisely, its lower bound) can be attributed to many possible trajectories between any two sets of APs, thus rendering unfeasible an accurate inference of the actual locations and trajectory. Moreover, as a distance also depends on the time difference between the location proofs, attributing a single distance to a given trajectory is even more challenging.

Adversary: AP operator(s). SecureRun is also designed to prevent curious AP operator(s) from tracking users' locations, particularly by linking activity proofs to reconstruct users' trajectories. Note that the need for physical proximity to interact with the APs makes it difficult to protect the users' privacy with respect to the AP operator(s). For this purpose, SecureRun relies on both randomized pseudonyms (generated by the service provider) and silence periods. On every sampling point, a new pseudonym is used, therefore it is difficult for AP operator(s) to link users' proofs and activities. For single-operator scenarios, the use of silence periods reduces the chances of an AP operator linking users' proofs during an activity. Note that, even if no silence periods are used, the operator can track a user only during her activity, without being able to link different activities over time. Thus, this prevents the AP operator from inferring patterns from activity trajectories over time. In addition, unlike the service provider, the operators have no personal information about the users (*e.g.,* their names). Note that SecureRun does not provide formal privacy guarrantees; it substantially improves the privacy of the users of activity-tracking applications in a best-effort fashion.

Finally, quantifying the location-privacy of users when pseudonyms and silence periods are employed is a typical mix-zone [58]/mix-network problem. Note, however, that in SecureRun the pseudonym-change strategy optimizes the sum of the distance proofs, whereas for traditional mix-zones it optimizes the users' privacy. In such situations, the location privacy of a user depends on the other users [59]–[61]: In general, privacy increases with the number of users.

8 Acknowledgments

The authors are extremely grateful to Sébastien Henri for helping implement SecureRun on a router.

[12] A similar approach has been proposed in systems such as PKIs [55] and anonymous credentials [56] to ensure *non-transferability*.

[13] Note that most OPE schemes leak some information about the plaintext, in addition to the numeric order [53]. However, recent proposals do not suffer from this problem [57].

9 Conclusion and Future Work

Activity-based social networks have become increasingly popular over the last few years. In their current form, such systems rely on the users' mobile devices to report the users' real locations while they pursue their activities. This provides neither security guarantees against cheaters, nor privacy protection against curious social-network providers, thus potentially threatening their adoption.

In this paper, we have proposed SecureRun, a system for providing secure and private proofs of location-based activities. SecureRun relies on the existing wireless access-point networks deployed in urban areas (at the cost of only a software upgrade, hence alleviating the need for deploying ad-hoc infrastructures), and it provides protection for both users and service providers. Our experimental evaluation, conducted using real data-sets of deployed wireless access-points and actual users' outdoor activities, shows that SecureRun achieves a good accuracy (up to 79%) when estimating a lower-bound of the distance that users cover during their activities, and it provides privacy and security properties. From a practical perspective, we envision our scheme to be of interest for strategic partnerships between social-network providers and access point network operators. We have focused our description and evaluation of SecureRun on distance summaries and have sketched a solution for elevation gain summaries as well. In addition, our proof-of-concept implementation of SecureRun on a router has shown that it can be deployed in practice. As such, this work constitutes a first step towards the design of secure and private activity-based social networks.

As part of future work, we consider (1) further improving SecureRun's accuracy by optimizing the unplanned sampling algorithms and (2) evaluating SecureRun on a real testbed of deployed access points to assess its technical feasibility and its performance in practice and (3) improving the users' location privacy, with respect to the AP operators, by *e.g.*, introducing dummy requests to confuse the adversary, using homomorphic encryption to combine LPs and (4) exploring the use of zero-knowledge proof techniques to prove the activity summaries without requiring the users to reveal their locations. Finally, to quantify the users' location privacy, we contemplate modeling the system (in the presence of many users pursuing location-based activities in the same region) as a mix-zone problem, define formal privacy metrics and evaluate them on real data-sets or through experiments.

References

[1] A. Pham, K. Huguenin, I. Bilogrevic, and J.-P. Hubaux, "Secure and private proofs for location-based activity summaries in urban areas," in *Proc. of ACM UbiComp*, 2014, pp. 751–762.

[2] "Collect points when running with the Helsana Trails App," http://www.helsana.ch/docs/Quick-guide-collect-points-when-running-iphone.pdf, last visited: Feb. 2015.

[3] "Achievemint," http://www.achievemint.com.

[4] "Higi," https://higi.com/.

[5] "Fitstudio," https://www.fitstudio.com/.

[6] "Oscar Health Using Misfit Wearables To Reward Fit Customers," http://www.forbes.com/sites/stevenbertoni/2014/12/08/oscar-health-using-misfit-wearables-to-reward-fit-customers/, last visited: Feb. 2015.

[7] "Swisscom ski cup," http://www.swisscom.ch/en/about/medien/press-releases/2013/10/20131028_MM_Swisscom_Snow_Cup.html, last visited: Feb. 2015.

[8] "Nike+ badges and trophies," http://www.garcard.com/nikeplus.php, last visited: Feb. 2015.

[9] "Wear this device so the boss knows you are losing weight," http://www.bloomberg.com/news/2014-08-21/wear-this-device-so-the-boss-knows-you-re-losing-weight.html, last visited: Feb. 2015.

[10] "Wearable tech is plugging into health insurance," http://www.forbes.com/sites/parmyolson/2014/06/19/wearable-tech-health-insurance/, last visited: Feb. 2015.

[11] "Your boss would like you to wear a jawbone fitness tracker," http://www.bloomberg.com/news/articles/2014-12-10/jawbone-up-for-groups-a-plan-to-get-employers-to-buy-fitness-bands, last visited: Feb. 2015.

[12] "Health insurer's app helps users track themselves," http://www.technologyreview.com/news/516176/health-insurers-app-helps-users-track-themselves/, 2013.

[13] M. Gruteser and B. Hoh, "On the Anonymity of Periodic Location Samples," in *Proc. PerCom*, 2005, pp. 179–192.

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

[15] Y. Matsuo, N. Okazaki, K. Izumi, Y. Nakamura, T. Nishimura, and K. Hasida, "Inferring long-term user property based on users," in *Proc. of IJCAI*, 2007, pp. 2159–2165.

[16] A. Noulas, M. Musolesi, M. Pontil, and C. Mascolo, "Inferring interests from mobility and social interactions," in *Proc. of NIPS Workshops*, 2009.

[17] D. Crandall, L. Backstrom, D. Cosley, S. Suri, D. Huttenlocher, and J. Kleinberg, "Inferring social ties from geographic coincidences," *Proc. of PNAS*, vol. 107, 2010.

[18] "Yes, those free health apps are sharing your data with other companies," http://www.theguardian.com/technology/appsblog/2013/sep/03/fitness-health-apps-sharing-data-insurance, last visited: Feb. 2015.

[19] "Strava, popular with cyclists and runners, wants to sell its data to urban planners," http://blogs.wsj.com/digits/2014/05/07/strava-popular-with-cyclists-and-runners-wants-to-sell-its-data-to-urban-planners/, last visited: Feb. 2015.

[20] B. Carbunar and R. Potharaju, "You unlocked the Mt. Everest badge on Foursquare! Countering location fraud in geosocial networks," in *Proc. of MASS*, 2012, pp. 182–190.

[21] M. Rahman, B. Carbunar, and U. Topkara, "A secure protocol for managing low power fitness trackers," *IEEE Trans. on Mobile Computing*, 2015, to appear.

[22] M. Naveed, X. Zhou, S. Demetriou, X. Wang, and C. A. Gunter, "Inside job: Understanding and mitigating the threat of external device mis-bonding on android," in *Proc. of NDSS*, 2014, pp. 23–26.

[23] W. He, X. Liu, and M. Ren, "Location cheating: A security challenge to location-based social network services," in *Proc. of IEEE ICDCS*, 2011, pp. 740–749.

[24] K. Zhang, W. Jeng, F. Fofie, K. Pelechrinis, and P. Krishnamurthy, "Towards reliable spatial information in LBSNs." in *Proc. of ACM UbiComp*, 2012, pp. 950–955.

[25] "Fake Track, Simulate GPS Routes," https://play.google.com/store/apps/details?id=com.trinus.faketrack&hl=en, last visited: Feb. 2015.

[26] "How to tell if someone used Digital Epo to cheat on Strava," http://www.scarletfire.co.uk/how-to-tell-if-someone-used-digital-epo-to-cheat-on-strava/, last visited: Feb. 2015..

[27] "Digitalepo," http://www.digitalepo.com/.

[28] "Fitbit cheat-o-matic," https://www.youtube.com/watch?v=3rrRFW0j5Vk, last visited: Feb. 2015.

[29] J. Brassil and P. K. Manadhata, "Proving the location of a mobile device user," in *2012 Virgina Tech Wireless Symposium*, 2012.

[30] W. Luo and U. Hengartner, "Veriplace: A privacy-aware location proof architecture," in *Proc. of GIS*, 2010, pp. 23–32.

[31] B. Carbunar, R. Sion, R. Potharaju, and M. Ehsan, "Private badges for geosocial networks," *IEEE Trans. on Mobile Computing*, vol. 13, no. 10, pp. 2382–2396, 2014.

[32] "Garmin connect," http://connect.garmin.com.

[33] "FON," https://corp.fon.com/en.

[34] I. Polakis, S. Volanis, E. Athanasopoulos, and E. P. Markatos, "The man who was there: validating check-ins in location-based services," in *Proc. of ACM ACSAC*, 2013, pp. 19–28.

[35] M. Talasila, R. Curtmola, and C. Borcea, "Link: Location verification through immediate neighbors knowledge," in *Proc. of MOBIQUITOUS*, 2012, pp. 210–223.

[36] Z. Zhu and G. Cao, "Toward privacy preserving and collusion resistance in a location proof updating system," *IEEE Trans. on Mobile Computing*, vol. 12, no. 1, pp. 51–64, 2013.

[37] S. Gambs, M.-O. Killijian, M. Roy, and M. Traoré, "PROPS: A privacy-preserving location proof system," in *Proc. of IEEE SRDS*, 2014, pp. 1–10.

[38] M. Jadliwala, S. Zhong, S. Upadhyaya, C. Qiao, and J.-P. Hubaux, "Secure distance-based localization in the presence of cheating beacon nodes," *IEEE Trans. on Mobile Computing*, vol. 9, no. 6, pp. 810–823, 2010.

[39] L. Hu and D. Evans, "Localization for mobile sensor networks," in *Proc. of ACM MobiCom*, 2004, pp. 45–57.

[40] J.-P. Sheu, W.-K. Hu, and J.-C. Lin, "Distributed localization scheme for mobile sensor networks," *IEEE Trans. on Mobile Computing*, vol. 9, no. 4, pp. 516–526, 2010.

[41] S. Saroiu and A. Wolman, "Enabling new mobile applications with location proofs," in *Proc. of HotMobile*, 2009, p. 3.

[42] D. Singelee and B. Preneel, "Location verification using secure distance bounding protocols," in *Proc. of IEEE MASS*, 2005, pp. 1–7.

[43] J. T. Chiang, J. J. Haas, and Y.-C. Hu, "Secure and precise location verification using distance bounding and simultaneous multilateration," in *Proc. of ACM WiSec*, 2009, pp. 181–192.

[44] S. Capkun and J.-P. Hubaux, "Secure positioning of wireless devices with application to sensor networks," in *Proc. of IEEE INFOCOM*, vol. 3, 2005, pp. 1917–1928.

[45] J. Reid, J. M. G. Nieto, T. Tang, and B. Senadji, "Detecting relay attacks with timing-based protocols," in *Proc. of ACM AsiaCCS*, 2007, pp. 204–213.

[46] A. Uchiyama, S. Fujii, K. Maeda, T. Umedu, H. Yamaguchi, and T. Higashino, "Upl: Opportunistic localization in urban districts," *IEEE Trans. on Mobile Computing*, vol. 12, no. 5, pp. 1009–1022, 2013.

[47] J. Koo, J. Yi, and H. Cha, "Localization in mobile ad hoc networks using cumulative route information," in *Proc. of ACM UbiComp*, 2008, pp. 124–133.

[48] T. Kohno, A. Broido, and K. C. Claffy, "Remote physical device fingerprinting," in *Proc. of IEEE S&P*, 2005, pp. 211–225.

[49] D. B. Faria and D. R. Cheriton, "Detecting identity-based attacks in wireless networks using signalprints," in *Proc. of ACM WiSec*, 2006, pp. 43–52.

[50] L. C. C. Desmond, C. C. Yuan, T. C. Pheng, and R. S. Lee, "Identifying unique devices through wireless fingerprinting," in *Proc. of ACM WiSec*, 2008, pp. 46–55.

[51] "Interior Point OPTimizer," https://projects.coin-or.org/Ipopt.

[52] A. Boldyreva, N. Chenette, Y. Lee, and A. O'Neill, "Order-preserving symmetric encryption," in *Proc. of EUROCRYPT*, 2009, pp. 224–241.

[53] A. Boldyreva, N. Chenette, and A. O'Neill, "Order-preserving encryption revisited: Improved security analysis and alternative solutions," in *Proc. of CRYPTO*, 2011, pp. 578–595.

[54] "Freewifi," http://www.free.fr/adsl/pages/internet/connexion/acces-hotspot-wifiFree.html, last visited: Jan. 2014.

[55] O. Goldreich, B. Pfitzman, and R. Rivest, "Self-delegation with controlled propagation—or—what if you lose your laptop," in *Proc. of CRYPTO*, 1998.

[56] J. Camenisch and A. Lysyanskaya, "An efficient system for non-transferable anonymous credentials with optional anonymity revocation," in *Proc. of EUROCRYPT*, 2001.

[57] R. Popa, F. Li, and N. Zeldovich, "An ideal-security protocol for order-preserving encoding," in *Proc. of S&P*, 2013, pp. 463–477.

[58] A. R. Beresford and F. Stajano, "Mix zones: User privacy in location-aware services," in *Proc. of IEEE PERCOM Workshops*, 2004, p. 127.

[59] L. Buttyán, T. Holczer, and I. Vajda, "On the effectiveness of changing pseudonyms to provide location privacy in vanets," in *Proc. of ESAS Workshop*, 2007, pp. 129–141.

[60] C. Diaz, S. Seys, J. Claessens, and B. Preneel, "Towards measuring anonymity," in *Proc. of PETS*, 2003, pp. 54–68.

[61] L. Bindschaedler, M. Jadliwala, I. Bilogrevic, I. Aad, P. Ginzboorg, V. Niemi, and J.-P. Hubaux, "Track me if you can: On the effectiveness of context-based identifier changes in deployed mobile networks." in *Proc. of NDSS*, 2012.

Anh Pham is a Ph.D. candidate at EPFL. She earned her M.Sc. degree in mobile computing and network security from KTH Royal Institute of Technology, Sweden, and Aalto University, Finland, in 2013. She was granted the Erasmus Mundus scholarship for the entirety of her master's studies. She earned her B.Sc. in information technology from Vietnam National University, Vietnam, in 2010. Her research interests include security and privacy in health-related applications and online social networks.

Kévin Huguenin is a permanent researcher at LAAS-CNRS, France, which he joined in 2014. Prior to that, he worked as a post-doctoral researcher at EPFL and at McGill University. He also collaborated with Nokia Research and he worked as an intern at Telefonica Research. He earned his Ph.D. in computer science from the Université de Rennes, France, in 2010 and his M.Sc. degree from École Normale Supérieure de Cachan and the Université de Nice – Sophia Antipolis, France, in 2007. His research interests include security and privacy in networks and distributed systems.

Igor Bilogrevic is a research scientist at Google, Switzerland, which he joined in 2014. He earned his Ph.D. on the privacy of context-aware mobile networks from EPFL in 2014. From 2010 until 2012, he worked in collaboration with the Nokia Research Center on privacy in pervasive mobile networks, encompassing social community and location privacy and information-sharing. In 2013, he worked as an intern at PARC (a Xerox Company) on topics related to private data analytics. His main research interests lie at the frontiers between privacy, security and user experience.

Italo Dacosta is a post-doctoral researcher at EPFL, which he joined in 2014. Prior to that, he worked as a post-doctoral researcher at KU Leuven, Belgium. He earned his Ph.D. in computer science and his M.Sc. degree in information security from the Georgia Institute of Technology, USA. He earned his B.Sc. degree in electronic and communication engineering from the Universidad de Panama, Panama, in 2002. He is also a former Fulbright grant recipient. His research interests include privacy enhancing technologies (PETs), identity management and network and mobile security.

Jean-Pierre Hubaux is a full professor at EPFL, which he joined in 1990. His current research activity is focused on privacy, notably in pervasive communication systems and online social networks. He has recently started research activity in genomic privacy, in close collaboration with geneticists. 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. Since 2007, he has been one of the seven commissioners of the Swiss FCC. He is a fellow of both the ACM and IEEE.