Secure and Private Proofs for Location-Based Activity Summaries in Urban Areas

Anh Pham, Kévin Huguenin, Igor Bilogrevic, and Jean-Pierre Hubaux
EPFL, Switzerland
{thivananh.pham,kevin.huguenin,igor.bilogrevic,jean-pierre.hubaux}@epfl.ch

ABSTRACT

Activity-based social networks, where people upload and share information about their location-based activities (*e.g.,* the routes of their activities), are increasingly popular. Such systems, however, raise privacy and security issues: the service providers know the exact locations of their users; the users can report fake location information to, for example, unduly brag about their performance. In this paper, we propose a secure privacy-preserving system for reporting location-based activity summaries (*e.g.,* the total distance covered and the elevation gain). Our solution 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 our solution by using real data-sets from the FON community networks and from the Garmin Connect activity-based social network, and show that it can achieve tight (up to a median accuracy of 79%) 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).

Author Keywords

Location privacy; Social networks; Location proofs

ACM Classification Keywords

C.2.0 Security and protection

INTRODUCTION

Over the last years, the presence and usage of embedded sensors in mobile devices has significantly increased. Location-based services (LBSs) are nowadays able to keep users informed about traffic conditions, significant events happening in proximity and nearby presence of other people with similar interests. More recently, LBSs are increasingly used by people to track, monitor and share their physical activities and performance over time; in particular, health- and wellness-related applications, such as Fitbit [10], Achievemint [1], Garmin connect [13], Nike+ [24] and Jawbone UP [20], allow users to keep track of their performance while running, hiking or cycling. In the current form of such systems, the users' mobile devices or activity trackers collect and send the users' locations (while pursuing their activities) to the service provider.

A popular feature of such applications is the ability to share summaries of users' activities or performance statistics with other users or service providers on social networks. For instance, users can share the total distance covered during their activities, the cumulative elevation gain and the actual path. In exchange for their data, users can be rewarded with coupons and discounts [32] or even with cash [1], with awards in competitions [25, 30], or simply with a better "social reputation" within their social circles.

Although activity tracking and sharing services are gaining popularity, there are two important issues that can hinder their wide-scale adoption and viability. First, users' location data, which is known to service providers, can be used to infer private information about them, such as their home/work locations [14, 17], activity preferences [23], interests [26] and social networks [8,22]. Second, users might be tempted to cheat when reporting their performance [6], in order to obtain a better reward, which can endanger the viability of the system for the service provider and its affiliates, as well as its attractiveness to other users. Location cheating can be achieved by making mobile devices report erroneous location information to the activity tracker app, or by spoofing the GPS or Wi-Fi signals used to geo-locate the users' [16].

A straightforward solution to those issues would consist in enforcing the use of either secure and/or privacy-preserving location proofs for users [4,16,21], 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), respectively. In fact, solutions guaranteeing property (1) would benefit the service provider by ensuring that cheating is infeasible, whereas solutions satisfying (2) would protect users' location privacy but would provide a too coarse-grained location resolution to be useful for the purposes of obtaining a reward or comparing performances.

In this paper, we propose a novel infrastructure-based approach that provides guarantees both in terms of cheating prevention 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 to-

UbiComp'14, September 13–17, 2014, Seattle, USA.

ACM XXX-X-XXXX-XXXX-X/14/09...$15.00.
http://dx.doi.org/10.1145/XXXXXXX.XXXXXXX

tal distance covered during an activity. Our approach relies on existing wireless access point (AP) networks, and it 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. Our approach 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 a user's 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. Our protocol produces, in a privacy-preserving way, a secure and accurate lower bound of the actual distance covered by a user while performing an activity. Finally, our solution is able to take advantage of the co-existence of multiple access point operators to improve the accuracy/privacy trade-off. 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 [13] 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 [11]. Moreover, to evaluate the benefits of having multiple operators in a given area, we extract the locations of a second network for the urban area of Paris. The experimental results show that our solution achieves a good accuracy (up to a median accuracy of 79%) and it can gracefully balance accuracy and privacy. We also conduct a sensitivity analysis to evaluate the effect of the distribution of the access points on the performance of our solution.

The remainder of the paper is organized as follows. We first survey the related work and we introduce the system and adversarial models. We then present our solution and we 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 this paper.

RELATED WORK

Prior works that study the secure verification of location information can, from a broad perspective, be grouped in two categories, depending on the presence or absence of the infrastructure. We first discuss the infrastructure-independent studies [19, 31, 33], and then we discuss the infrastructure-dependent ones [4, 15, 21, 28].

In the infrastructure-independent approach, a user obtains location evidences from her neighbors by using short range communication technologies, such as Bluetooth [31, 33]. Specifically, Talasila et al. [31] propose a location authentication protocol called LINK (Location verification through Immediate Neighbors Knowledge), 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. [33] propose the APPLAUS system, where mutually co-located users rely on Bluetooth communications to generate their location claims, which are then sent to a centralized location verifier. In addition to the security and privacy guarantees presented in [31], Zhu et al. [33] allow individual users to evaluate their own location privacy and decide whether to accept location proof requests by other users. Jadliwala et al. [19] provide a formal analysis of the conditions that need to be satisfied in an ad-hoc network, in order to enable the existence of any distance-based localization protocols in wireless networks.

More in line with our work, the infrastructure-dependent studies assume the presence of a centrally-operated set of access points (AP) to produce and verify location claims. For instance, to ensure the presence of a user in a given region, the AP can require her to be 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 [28], or any distance bounding protocol [5, 7, 29], which enables the AP to check the minimum distance between itself and the user. In particular, [5] 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, which is a document signed by the witnesses to certify that at a specific time, the user is at a specific geographical location [28]; for example, an AP can embed its coverage range, its center coordinate and a timestamp in the location proof, in order to certify that at the specified timestamp, the user is in the coverage area of the AP. Alternatively, in [21] a user can choose to obtain location proofs for different levels of granularity for the precision of her location, and choose the one to disclose to the service provider depending on her preferences and privacy sensitivity.

He et al. [16] present a study that deals specifically with a cheating attack on a social network (Foursquare). The authors show how the users can easily override or bypass the GPS verification mechanisms of the service provider by, notably, modifying the values that are returned by the API calls to the geo-location interface of the smartphones. The attacker can achieve such a result by either using the APIs provided by the online services, or via device emulators.

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

SYSTEM ARCHITECTURE

In this section, we describe the different entities involved in our system: A user, a Wi-Fi network operator and a social network provider. Figure 1 depicts the system we consider and a sketch of the solution. We also describe the adversarial model in this scenario.

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). Therefore, 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. 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. A pseudonym contains a pair of public/private keys, generated with a public-key encryption scheme such as RSA [27] or Elgamal [9]. 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 allowing us to prevent proxy/relay attacks [16] by using delay-based challenge-response mechanisms.

Users might be tempted to cheat by reporting locations that are different from their actual locations, in order to unduly brag about their performance or obtain rewards. To do so, users might, for instance, forge messages or reuse messages they, or their friends, obtained in the past.

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. 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, whereas GK_{priv} is only known to the network operator and to its APs.

The access point operators are interested in tracking the users' locations, based on the information obtained by all of their APs. They are assumed to be *semi-honest* or *honest-but-curious*, meaning that they do not deviate from the protocol specified in our solution but they simply analyze the information they collect while executing the protocol. We further assume that different network operators do not collude with each other and that they do not collude with the social network provider.

Social network provider

We assume that there is a social network provider that offers activity summaries and sharing services for 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 is interested in the users' locations and it is assumed to be *honest-but-curious*.

SOLUTION

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

Overview

From a general perspective, our solution 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).[1] The times/positions at which users request such location proofs are determined by a *sampling algorithm*.[2]

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 are free from 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 collected, while pursuing her activity, to the social network provider that performs the adequate verifications; if they are successful, 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' location (essentially unlinkability of activity proofs) with respect to the access point operators; the lack of location information in activity proofs provides protection with respect to

[1] 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 between two points is a good approximation of the Haversine distance between the corresponding locations.

[2] For the sake of clarity, we describe the sampling algorithm after the location and activity proofs.

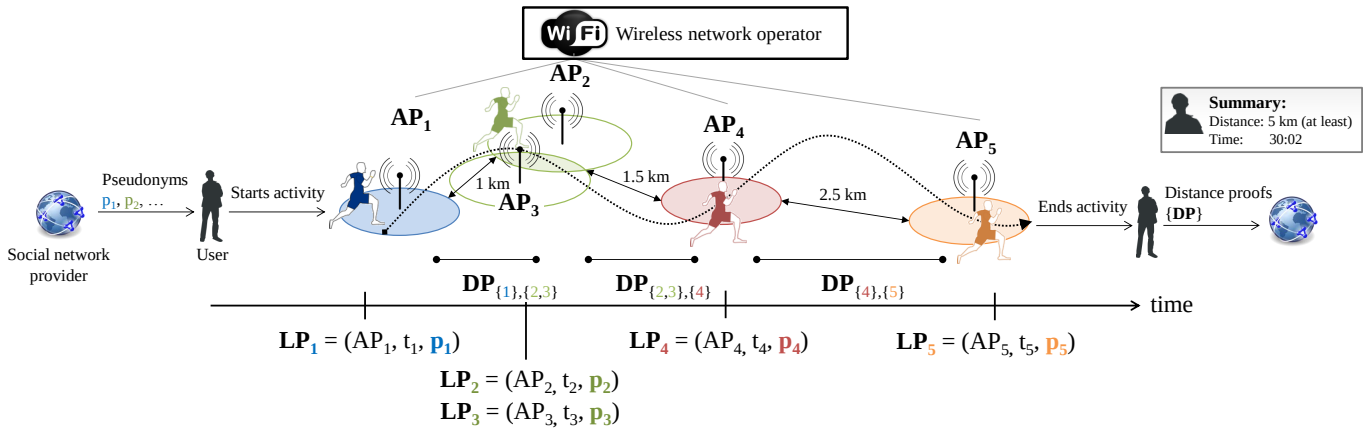


Figure 1. System architecture of the proposed solution. A user first obtains a set of pseudonyms $\{p_1, \ldots, p_K\}$ from the social network provider. Then, while performing a location-based activity along the dotted trajectory, she sporadically requests location proofs (LP) at times t_i, using pseudonyms 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.

the social network provider. Finally, the use of digital signatures and pseudonyms, combined with the fact that the activity proofs represent lower bounds of the user's actual performance, provide security properties with respect to dishonest users.

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}, namely 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 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 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 $\mathcal{C}((x, y), R)$.

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 she collected at two consecutive sampling times t_i and t_{i+1}. The contacted access point first combines the different location proofs, 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 a time t_i from the following expression

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

The access point repeats the same operation for the location proofs obtained at sample time $i + 1$.

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

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

A tight approximation of d_i can be obtained by using a non-linear optimization toolbox such as IPOPT [18].

With respect to the elevation proofs, the following expression gives a lower bound of the cumulative elevation gain[3]

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

achieved by a user during the time interval $[t_i, t_{i+1}]$.

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

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 2 illustrates the different stages of the generation of activity proofs in the case of the covered distance and of the elevation gain.

Finally, the access point generates an activity proof $\mathrm{sig}_{GK_{\mathrm{priv}}}\{d_i, e_i, [t_i, t_{i+1}], \{P_i, P_{i+1}\}\}$ and sends it back.

Activity summary

To publish an activity summary on her profile, a 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 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.

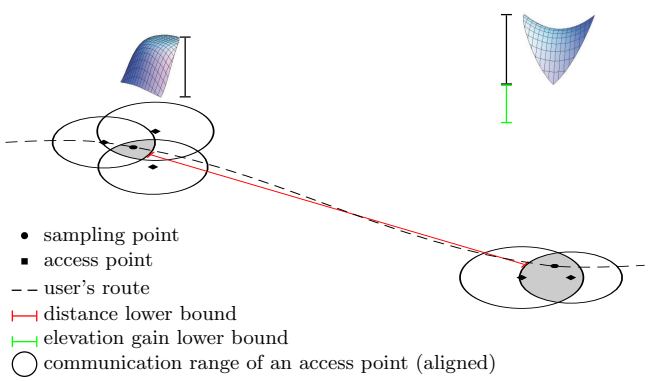


Figure 2. Computation of distance and elevation proofs. The shaded areas correspond to the intersections of the location proofs obtained at the same sampling time. The 3D plots correspond to the elevation profiles of the shaded areas, based on which the lower-bound of the elevation gains are computed.

Sampling algorithms

We now describe our sampling algorithm. The sampling algorithm determines the times/positions (namely the sampling times/points) 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 a 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 of the activity she is about to start, 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 tracker applications (including Garmin's) in order to enable users to *compete* against their own previous performance. For instance, the activity tracker 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 one single access point operator, and subsequently multiple such operators.

We focus on the case of distance proofs[4]. 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 locations 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 locations 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 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 3. The duration ΔT of the silence period[5] 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

[4]The problem is simpler for elevation proofs as the optimal sampling strategy simply consists in requesting location proofs at the points where the elevation is a locally minimal/maximal.

[5]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.

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.

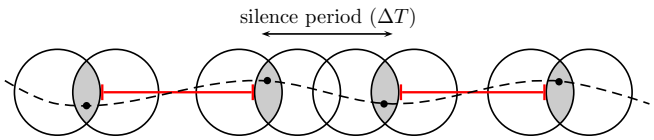


Figure 3. Silence period. By implementing a silence period between every pair of successive distance proofs (*i.e.,* not requesting a distance proof for this period), a user reduces the risk of her distance proofs being linked by the access point, hence protecting her privacy.

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. At every sample point, a user requests location proofs from the access points of all the operators. Then, for each interval between two successive sampling points, she determines which operator would provide the largest distance proof, by computing locally the lower-bound distance for the location proofs she collected, and requests a distance proof from an access point that belongs to this operator. In order to protect her privacy, a user never requests two successive distance proofs from the same operator, unless she implements a silence period. With two operators, a user alternatively requests distance proofs from each of the two access point operators, as illustrated in Figure 4.

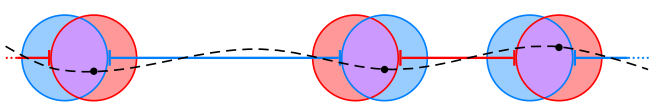


Figure 4. Case of multiple access point operators (Operator 1 in blue and Operator 2 in red). At every sampling point, a user requests location proofs from both operators. Then, she requests distance proofs alternatively from different operators to reduce the risk of linking the distance proofs she collects without reducing the accuracy of her activity summary (unlike when implementing 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

As the path and the location of the access points are known to the algorithm, a user can determine in advance the location proofs she can collect (and the resulting areas $\{A_i\}$, as defined in Eq. (1)) at all the points on the path. We sample regularly on the path (*e.g.,* every 10 m) and we process each sample in a greedy fashion. The first two points of the path are, by default, sampling points. We iterate on the points of the path, starting at the third one. We process point i as follows: (1) We add point i to the set of sampling points, and (2) we remove the last recently added sampling point, if this yields a larger lower-bound distance. Algorithm 1 embodies a pseudo-code version of the planned sampling algorithm, where $A(p)$ denotes the area resulting from the combination of the locations proofs the users collects at a point p and $d(\cdot,\cdot)$ denotes the minimum distance between two such areas.

Note that in practice, a user would 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).

Algorithm 1 Planned sampling algorithm.

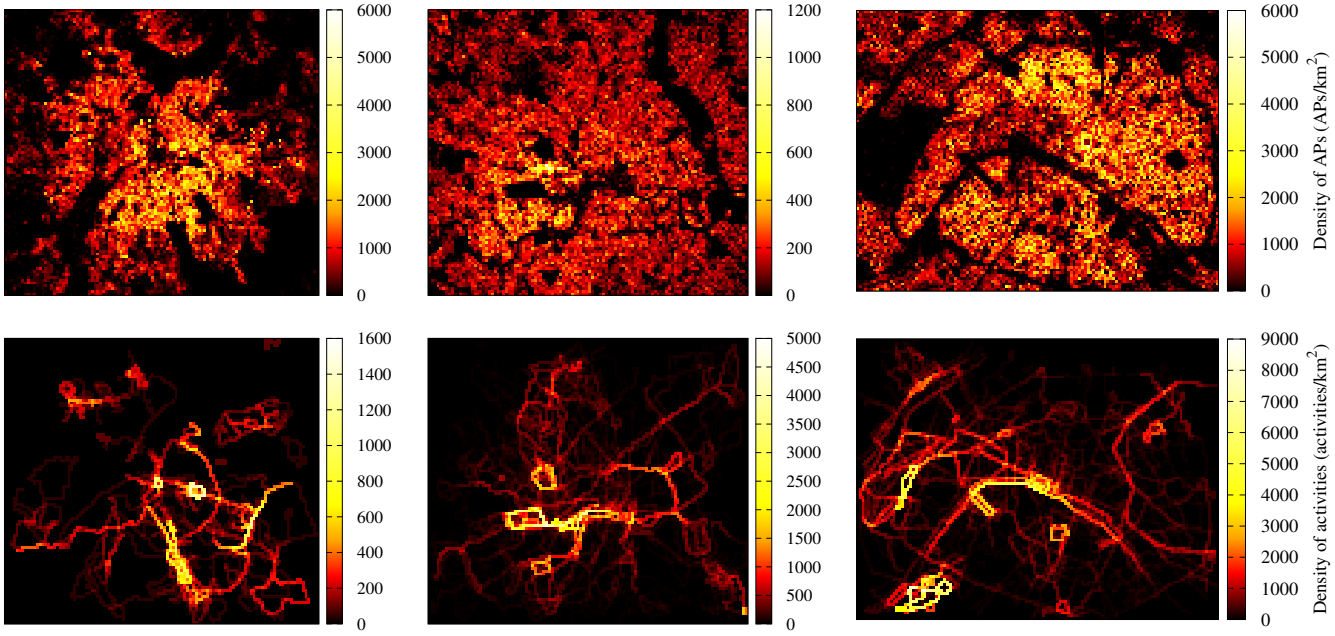
Input: $(p_1, \ldots, p_n)$ ▷ Sequence of points on the path
Output: S ▷ Set of sampling points
$a \leftarrow 1$ ▷ Index of the next-to-last sampling point
$b \leftarrow 2$ ▷ Index of the last sampling point
$S \leftarrow \{a, b\}$
for $i = 3$ **to** n **do**
  if $d(A(p_a), A(p_i)) > d(A(p_a), A(p_b)) + d(A(p_b), A(p_i))$ **then**
    $S \leftarrow S - \{b\}$
  else
    $a \leftarrow b$
  end if
  $S \leftarrow S \cup \{i\}$
  $b \leftarrow i$
end for

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 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.

Summary

In this section, we have presented a solution for providing secure and privacy-preserving activity summaries, and we 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 our solution are due to the fact that (1) the distances covered inside the areas $\{A_i\}$ as well as 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 report on the evaluation of the accuracy of our solution in the next section. The security and the privacy properties of our solution 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 the "Security and privacy analysis" section.



(a) Brussels (b) London (c) Paris

Figure 5. Heat-maps of the densities of FON access points (top row) and of Garmin Connect activities (bottom row) in (a) Brussels, (b) London, and (c) Paris. Note that, for the sake of presentation, the color range differs from one map to another.

PERFORMANCE EVALUATION

We evaluate the performance of the proposed solution on real traces of users' activities from Garmin Connect [13], pursued in cities where wireless access points networks are deployed by the FON operator [11] (and possibly Free [12]). We consider scenarios where mobile users, equipped with Wi-Fi enabled devices, want to report the cumulative elevation gain and the total distance covered during their location-based activities (e.g., running). We focus our evaluation on three geographical areas corresponding to the cities of Brussels, London and Paris.

Data-sets

In order to evaluate our solution, we collected data-sets of access points locations and activities and we relied on the Google Elevation API. Table 2 contains general statistics about the (filtered) data-sets.

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 12 million hotspots worldwide, most of them 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 our solution. Overall, we obtained the locations of 92,280 unique APs[6] in Brussels, 39,776 unique APs in London, and 87,521 unique APs in Paris. In order to evaluate our solution 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$\pm$381 AP/km^2. Figure 5 (top) depicts the heat-maps of the densities of FON access points. 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.,* access points 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 in chunks of 500 m, and we determine for each chunk whether it is covered by

[6]We filtered out duplicated APs (that either have the same identifier or the exact same coordinates as another AP).

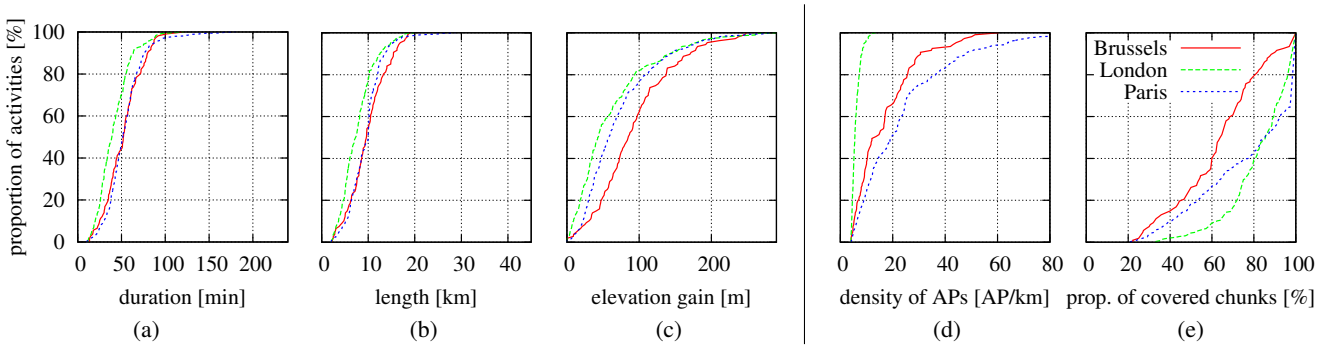


Figure 6. Experimental CDF of the (a) duration, (b) length, (c) elevation gain (d) density of AP (along the activity) and (e) proportion of covered chunks, among the activities from the Garmin data-set.

at least one access point (i.e., it intersects with the communication range of at least one access point). This metric is crucial for our solution to work as 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 dataset, 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 different filters applied to our raw data-set.

	Filter
Duration	$<$10 min or $>$ 4 h
Length	$<$2 km or $>$ 45 km
Inter-sample times	$>$ 10 min
Density of AP along activities	$<$ 4 AP/km
Proportion of covered chunks	$<$ 20%

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

Figure 6 shows the experimental cumulative distribution functions of the main characteristics of the activities used in our evaluation and Figure 5 (bottom) 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 observed a median inter-sample time of 3-4 seconds (which correspond to 7-11 meters).

Table 2 summarizes some relevant (with respect to our solution) statistics on the filtered data. It can be observed that the density of access points is lower in London but they are more uniformly spread, especially along activities (as illustrated by the relatively small standard deviation compared to Brussels and Paris). Consequently, the number of covered chunks is higher in London, thus letting us anticipate better results for our solution in London.

Elevation

In order to determine the minimum and maximum elevation of a given region, typically the intersection of discs centered at the AP locations (as required in our solution to compute lower-bounds of the elevation gains), we rely on the Google Elevation API: We pick, uniformly at random, 20 locations inside the region of interest, we query their elevation, and we compute the minimum and maximum values. We also use this API to compute the cumulative elevation gains of the activities extracted from Garmin Connect.

	Brussels	London	Paris
Number of AP	92,280	39,776	87,521
Number of activities	107	294	437
Density of AP (AP/km^2)	401$\pm$569	109$\pm$96.6	646$\pm$686
Density of AP along activities (AP/km)	17.1$\pm$12.0	5.99$\pm$1.67	23.8$\pm$18.6
Proportion of covered chunks (%)	63.9$\pm$20.0	83.0$\pm$15.0	77.7$\pm$23.5

Table 2. Summary of the statistics of the filtered data-sets (FON and Garmin Connect) used in the evaluation (mean and standard deviation).

Methodology

We implement our solution in a Java simulator and evaluate its performance for the activities from the Garmin Connect dataset, with the access point networks from the FON and Free data-sets (under the unit-disc communication model with a radius of 25 meters). For each activity, we simulate the execution of our solution 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 performance of our solution 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. As the summaries are lower bounds of the actual performance of the users, the accuracy is between 0 and 100%. We only report on the evaluation of the accuracy of distance summaries.

Results

First, we look at the absolute performance of our solution in different settings. Figure 8 shows a box-plot representation (first quartile, median, third quartile, and outliers) of the accuracy of our solution 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 our solution with two access point operators

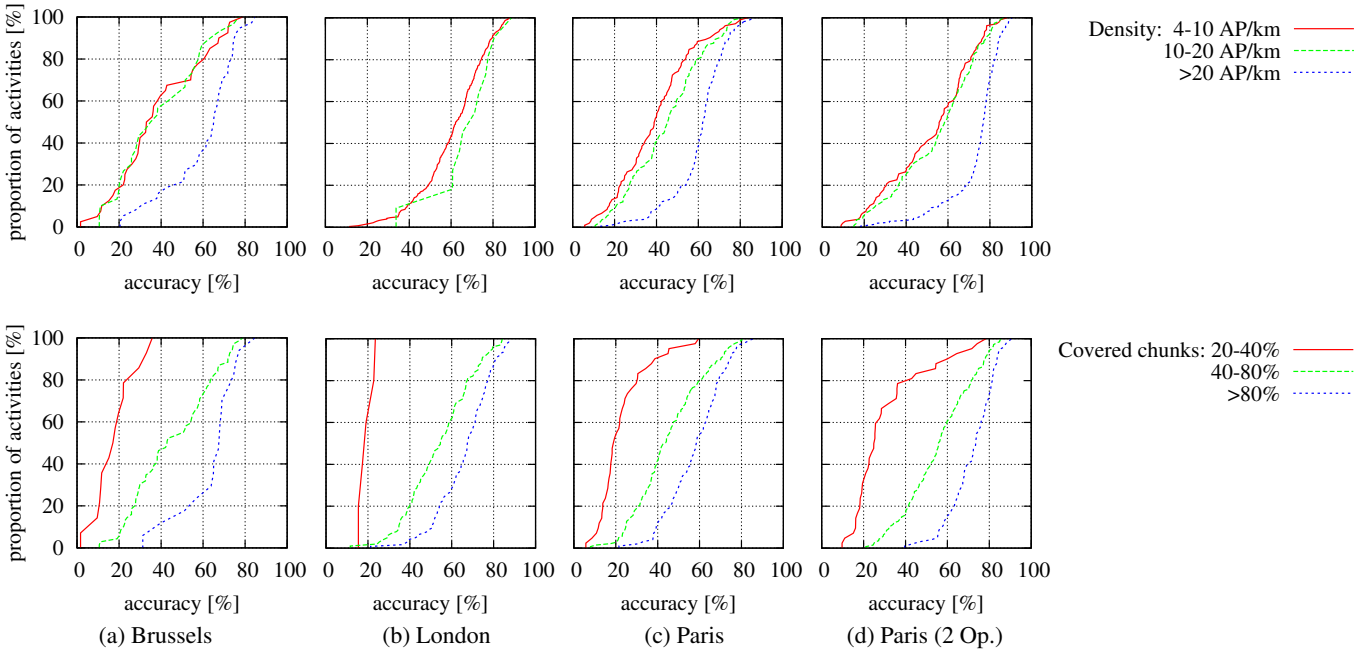
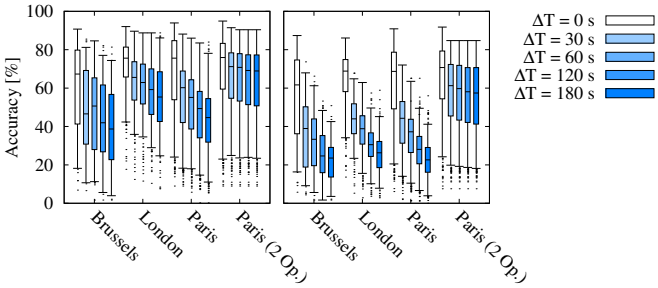


Figure 7. Sensitivity analysis of the accuracy, with respect to the density of access point along the activities (top) and to the proportion of covered chunks (bottom). The planned sampling algorithm was used, with silence periods of $\Delta T = 60$ s. Note that in London, all activities have a density $\leq$20 AP/km.

(Free and FON). Overall, our solution achieves good performance: up to a median accuracy of 75.6% (Paris, 1 operator, planned sampling, $\Delta T = 0$). This value drops to 68.7% 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 only slightly better (75.9%) 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 (68.9% for the case of two operators vs. 44.6% 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 the performance of our solution increases with the proportion of covered chunks.

To further study the sensitivity of our solution 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 in 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



(a) planned sampling (b) unplanned sampling

Figure 8. Accuracy of the distance summaries, with the (a) planned and (b) unplanned sampling algorithm, for different values of the duration of the silence periods, with the FON netwrok (+ Free for Paris 2 Op.).

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 7, 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 64.9% for activities with high densities, whereas it is only 50.6% for all the activities. Note that even for some activities with a high density, the accuracy can be quite low (*i.e.,* <20%). We investigated this issue by manually inspecting the paths of these activities; we found that, for example, there are activities where the user first runs to a stadium through a residential area and then runs a dozen of times inside the stadium on the 400-meter running track. Because the stadium is covered by

a single access point, all the chunks of the activity are covered. However, this is still not sufficient to obtain non-zero distance proofs, as all the location proofs inside the stadium are obtained from the same access point. Because the activity begins in a residential area with a high access point density, the average density over the complete activity is higher than 20 AP/km.

SECURITY AND PRIVACY ANALYSIS

In this section, we discuss the security and privacy properties provided by our mechanism, by considering three possible adversaries: the users, the service provider and the AP operator(s).

Adversary: User

First of all, we prevent users from forging location or activity proofs by using digital signatures. Moreover, valid location proofs can only be obtained if the users are in communication range with the APs.

Second, proxy attacks, in which two or more users collude in order to obtain valid location proofs, can be limited by introducing constraints on the execution time of the protocol; for example, the AP operator could impose a communication delay on the Wi-Fi interface that is smaller than the one achieved by connecting through the cellular network.

Third, users cannot double count some of the distances they cover because each activity proof contains the initial and final time instants. Hence, the service provider can check that they do not overlap before summing them up.

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

Adversary: Service provider

In our mechanism, the service provider has only access to location proofs and pseudonyms of the users. As location proofs do not contain any location information, it cannot link the distance proofs to actual locations. 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, hence 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.

In order to hide the time at which the distance proofs are obtained (in addition to their locations), a method based on order-preserving encryption [3] can be used. This would enable the service provider to check that the time intervals of the activity proofs are indeed disjoint, without knowing the actual time intervals, hence further protecting the privacy of the users.

Adversary: AP operator(s)

To prevent the AP operator(s) from tracking the locations of the users, notably by linking activity proofs in order to reconstruct the users' trajectories, our mechanism employs both randomized pseudonyms (generated by the service provider) as well as silence periods. Quantifying the location-privacy of users when pseudonyms and silence periods are employed is a typical mix-zone problem [2]. In such situations, the location privacy of a user depends on the other users as well, where the higher the number of users is, the better their privacy is. Note that, even if no silence periods are used (in the single operator scenario), the operator can only track a user 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. Note that, unlike the service provider, the operators have no personal information about the users (such as their names).

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 collect and to report the users' actual 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 wide-scale adoption.

In this paper, we propose a solution for providing secure and private proofs of location-based activities. Our solution relies on the existing wireless access point networks (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. By targeting activities pursued in urban areas, it does not require users to cooperate or exchange messages with each others in an ad-hoc manner. Our experimental evaluation, conducted using real data-sets of deployed wireless access points and actual users' outdoor activities, shows that our solutions achieves a good accuracy (up to 79%) when estimating a lower-bound of the distance that users cover during their activities, while providing 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 focused our description and evaluation of our solution on distance summaries and sketched a solution for elevation gain summaries as well. 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 plan to (1) further improve the accuracy of our solution by optimizing the sampling algorithms, (2) extend our evaluation to include the case of cumulative elevation gain summaries, and (3) evaluate our solution on a real testbed of deployed access points to assess its technical feasibility and its performance in practice. Finally, we plan to formalize the system (in the presence of multiple users pursuing location-based activities in the same region) as a mix-zone problem in order to quantify the loss of users' location privacy.

REFERENCES

1. Achievemint. `http://www.achievemint.com`. Last visited: Jan. 2014.
2. Beresford, A. R., and Stajano, F. Mix zones: User privacy in location-aware services. In *PERCOMW'04: Proc. of the 2nd IEEE Annual Conference on Pervasive Computing and Communications Workshops* (2004), 127–.
3. Boldyreva, A., Chenette, N., and O'Neill, A. Order-preserving encryption revisited: Improved security analysis and alternative solutions. In *CRYPTO'11: Proc. of the 31st Annual International Cryptology Conference* (2011), 578–595.
4. Brassil, J., and Manadhata, P. K. Proving the location of a mobile device user. In *2012 Virgina Tech Wireless Symposium* (2012).
5. Capkun, S., and Hubaux, J.-P. Secure positioning of wireless devices with application to sensor networks. In *INFOCOM '05. Annual Joint Conference of the IEEE Computer and Communications Societies*, vol. 3 (2005), 1917–1928.
6. Carbunar, B., and Potharaju, R. You unlocked the mt. everest badge on foursquare! countering location fraud in geosocial networks. In *Mobile Adhoc and Sensor Systems (MASS), 2012 IEEE 9th International Conference on*, IEEE (2012), 182–190.
7. Chiang, J. T., Haas, J. J., and Hu, Y.-C. Secure and precise location verification using distance bounding and simultaneous multilateration. In *Proc. of the second ACM conference on Wireless network security* (2009), 181–192.
8. Crandall, D., Backstrom, L., Cosley, D., Suri, S., Huttenlocher, D., and Kleinberg, J. Inferring social ties from geographic coincidences. *National Academy of Sciences 107* (2010).
9. ElGamal, T. A public key cryptosystem and a signature scheme based on discrete logarithms. In *Advances in Cryptology* (1985), 10–18.
10. Fitbit. `http://www.fitbit.com/de`. Last visited: Jan. 2014.
11. Fon. `https://corp.fon.com/en`. Last visited: Feb. 2014.
12. Freewifi. `http://www.free.fr/adsl/pages/internet/connexion/acces-hotspot-wifiFree.html`. Last visited: Jan. 2014.
13. Garmin connect. `http://connect.garmin.com`. Last visited: Jan. 2014.
14. Gruteser, M., and Hoh, B. On the Anonymity of Periodic Location Samples. In *International Conference on Security in Pervasive Computing* (2005).
15. Hasan, R., and Burns, R. Where have you been? secure location provenance for mobile devices. *arXiv preprint arXiv:1107.1821* (2011).
16. He, W., Liu, X., and Ren, M. Location cheating: A security challenge to location-based social network services. In *ICDCS '11: International Conference on Distributed Computing Systems (ICDCS)* (2011), 740–749.
17. Hoh, B., Gruteser, M., Xiong, H., and Alrabady, A. Enhancing Security and Privacy in Traffic-Monitoring Systems. *IEEE Pervasive Computing 5* (2006), 38–46.
18. Interior point optimzer. `https://projects.coin-or.org/Ipopt`. Last visited: Jan. 2014.
19. Jadliwala, M., Zhong, S., Upadhyaya, S., Qiao, C., and Hubaux, J.-P. Secure distance-based localization in the presence of cheating beacon nodes. *Mobile Computing, IEEE Transactions on 9*, 6 (2010), 810–823.
20. Jawbone up. `https://jawbone.com/up`. Last visited: Jan. 2014.
21. Luo, W., and Hengartner, U. Veriplace: A privacy-aware location proof architecture. In *Proc. of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems (GIS'10)* (2010), 23–32.
22. Mardenfeld, S., Boston, D., Pan, S., Jones, Q., Iamntichi, A., and Borcea, C. Gdc: Group discovery using co-location traces. In *International Conference on Social Computing* (2010).
23. Matsuo, Y., Okazaki, N., Izumi, K., Nakamura, Y., Nishimura, T., and Hasida, K. Inferring Long-term User Property based on Users. In *International Joint Conference on Artificial Intelligence* (2007).
24. Nike+ fuelband. `https://secure-nikeplus.nike.com/plus/`. Last visited: Feb. 2014.
25. Nike+ badges and trophies. `http://www.garcard.com/nikeplus.php`. Last visited: Feb. 2014.
26. Noulas, A., Musolesi, M., Pontil, M., and Mascolo, C. Inferring interests from mobility and social interactions. In *NIPS Workshop on Analyzing Networks and Learning with Graphs* (2009).
27. Rivest, R. L., Shamir, A., and Adleman, L. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM 21*, 2 (1978), 120–126.
28. Saroiu, S., and Wolman, A. Enabling new mobile applications with location proofs. In *Proc. of the 10th workshop on Mobile Computing Systems and Applications*, ACM (2009), 3.
29. Singelee, D., and Preneel, B. Location verification using secure distance bounding protocols. In *IEEE Conference on Mobile Adhoc and Sensor Systems Conference* (2005).

30. Swisscom ski cup. **http://www.swisscom.ch/en/about/medien/press-releases/2013/10/20131028_MM_Swisscom_Snow_Cup.html**. Last visited: Feb. 2014.

31. Talasila, M., Curtmola, R., and Borcea, C. Link: Location verification through immediate neighbors knowledge. In *Mobile and Ubiquitous Systems: Computing, Networking, and Services*. 2012, 210–223.

32. Walgreens steps with balance rewards. **https://www.walgreens.com/steps/stepslanding.jsp?ec=health_info_flyer_steps**. Last visited: Feb. 2014.

33. Zhu, Z., and Cao, G. Toward privacy preserving and collusion resistance in a location proof updating system. *Mobile Computing, IEEE Transactions on 12*, 1 (2013), 51–64.