2011 Eighth IEEE International Conference on Mobile Ad-Hoc and Sensor Systems

Collaborative Location Privacy

[†]Reza Shokri, [‡]Panos Papadimitratos, [†]George Theodorakopoulos, and [†]Jean-Pierre Hubaux
[†]LCA, EPFL, Lausanne, Switzerland, [‡]KTH, Stockholm, Sweden
[†]firstname.lastname@epfl.ch, [‡]papadim@kth.se

***Abstract*—Location-aware smart phones support various location-based services (LBSs): users query the LBS server and learn on the fly about their surroundings. However, such queries give away private information, enabling the LBS to identify and track users. We address this problem by proposing the first, to the best of our knowledge, user-collaborative privacy preserving approach for LBSs. Our solution, MobiCrowd, is simple to implement, it does not require changing the LBS server architecture, and it does not assume third party privacy-protection servers; still, MobiCrowd significantly improves user location-privacy. The gain stems from the collaboration of MobiCrowd-ready mobile devices: they keep their context information in a buffer, until it expires, and they pass it to other users seeking such information. Essentially, the LBS does not need to be contacted unless all the collaborative peers in the vicinity lack the sought information. Hence, the user can remain hidden from the server, unless it absolutely needs to expose herself through a query. Our results show that MobiCrowd hides a high fraction of location-based queries, thus significantly enhancing user location-privacy. To study the effects of various parameters, such as the collaboration level and contact rate between mobile users, we develop an epidemic model. Our simulations with real mobility datasets corroborate our model-based findings. Finally, our implementation of MobiCrowd on Nokia platforms indicates that it is lightweight and the collaboration cost is negligible.**

I. Introduction

Smart phones, among other increasingly powerful mobile computing devices, offer various methods of localization. Integrated GPS receivers or positioning services based on nearby communication infrastructure enable users to position themselves fairly accurately. This gives rise to a range of *Location-Based Services* (LBSs): users can query an LBS server and obtain information relevant to their current location and surroundings, that is, contextual data about specific points of interest. The value of LBSs is exactly in obtaining accurate and up-to-date information on the fly.

The flip-side of getting on-site high-quality on-demand information is the loss of users' privacy: Each time an LBS query is submitted, private information is revealed. The user can be linked to her location, and multiple pieces of such information can be linked together; thus, the profiling of users becomes possible. Clearly, the user could forgo the LBS benefits; e.g., she could download a large data volume and then search locally about specific context information. But this would be cumbersome, if not impractical, and it would be inefficient for obtaining information that changes dynamically over time.

In order to obtain as much information as possible about the LBS users, which will be mainly used for sending targeted advertisement to the users, the service providers track users over time using various techniques. For example, the service provider can explicitly ask for the users' contact information. However, even if the LBS does not perform any explicit user identification, it is still possible to finger-print users of specific applications [10], or de-anonymize them (i.e., infer their identity) by using their IP addresses or location [23], and then trace their whereabouts.

More importantly, independently of whether the user is identified or not, placing too much trust in LBS providers is undesirable. Indeed, the LBS operators may be tempted to misuse the rich data they collect, or they may, as opposed to cellular operators (who have a contract with their users), share the data with third-party companies that offer, for example, targeted advertisements. Moreover, the LBS data repositories may be targeted by attackers, who break into the LBS servers and obtain logs of user queries. The result in all cases is the same: user-sensitive data fall in the hands of untrusted parties.

Tracking the user over time and space, and then identifying her, implies not only loss of privacy for the user but possibly other dire consequences such as *absence disclosure*: learning that a user is away from her home could allow a house break-in or blackmail [3]. As a result, the need to enhance privacy for LBS users has been understood and several solutions have been proposed. One approach could be to blur the location information, e.g., by having the user's smart phone (or the privacy proxy) submit inaccurate samples to the LBS server. However, obfuscation approaches (e.g., spatial/temporal cloaking introduced in [16]) which can protect user location-privacy, degrade the user experience if users need high privacy: e.g., LBS responses would be inaccurate or untimely. Moreover, obfuscation cannot be effective against absence disclosure [29]. Another approach could be to introduce a third party in the system, acting between the user and the LBS: its role would be

978-0-7695-4469-4/11 $26.00 © 2011 IEEE
DOI 10.1109/MASS.2011.55

to protect the users' privacy. Such an intermediary proxy server, between the user and the LBS, could anonymize (and obfuscate) queries by removing any information that identifies the user or her device [13], [25]. Or it could blend one query with those of other users, so that the LBS server always sees a group of queries [24]. However, such approaches only shift the problem: the threat of an untrustworthy LBS server is addressed by the introduction of a new third-party server. Some other approaches require the LBS to change its operation, for example, by mandating it to process modified queries (submitted in different forms than actual queries of the user), or that it needs to store data differently (e.g., encrypted or encoded, to allow private access [14]).

Any such centralized intervention or any substantial changes to the LBS operation would be hard to adopt, simply because the LBS providers would have little incentive to fundamentally change their operation. Misaligned incentives have been identified as the root of many security problems [6]. Additionally, new proxy servers become as attractive for attackers as centralized LBSs. Hence, the lack of incentives and guarantees for protecting the users' location information, make these approaches infeasible in practice.

In order to enhance the location privacy of LBS users without any of the above-mentioned limitations, we propose here a new *user-centric* scheme. Mobile users concerned about their location privacy are indeed the most motivated entities to engage in protecting themselves. Our solution, called MobiCrowd, takes advantage of this fact, making the privacy-sensitive users responsible for their own privacy protection. Our approach requires no change of the LBS server architecture and its normal operation, it makes no assumption on the trustworthiness of the LBS or any other third-party server, and it enhances the privacy of mobile users in terms of both presence and absence disclosure.

MobiCrowd achieves this improvement thanks to a novel *collaborative privacy-protection mechanism*: basically, a user can avoid disclosing her location information, to the LBS server, if her device can have its LBS queries answered by nearby peers (i.e., other reachable user devices) that happen to have the sought data. Clearly, MobiCrowd would be most effective when there are many peers gathered at the same location. Indeed, this clustering phenomenon has been observed in human mobility studies [27]. Moreover, the places where people gather are points of interest, where users are most likely to ask an LBS for information. So, MobiCrowd would be used *exactly* where it is most *effective*.

We analyze our scheme experimentally and analytically, proposing an epidemic model for the dynamics of information sharing among users. The model captures the effect of many users clustering at the same place, and it can be used to test various "what-if" scenarios about MobiCrowd. This is a novel approach to evaluate a location-privacy preserving mechanism for mobile networks: it acts on the *parameters of their mobility model* rather than on some specific location traces. Thus, we can study the effects of a mixture of parameters and we can also identify the causes of high or low location privacy in various settings. We then perform a simulation on real mobility traces, and we show that the conclusions from the experimental evaluation verify the results derived from our model.

The threat of local observers sniffing the wireless channel trying to infer users' private information, is out of the scope of this paper; such a threat could exist with or without MobiCrowd and it can be alleviated by frequently changing device identifiers (e.g., changing MAC addresses for WiFi networks [18] similar to changing TMSI for GSM networks [5]). More importantly, local observers would have a tedious task and still be ineffective in collecting information: they would need to be physically present next to any given victim user, over long periods and across different locations. In contrast, a centralized LBS can by default observe *all* the queries of a user, which is why we focus on this much greater threat in this paper. However, in order to secure the scheme against untrustworthy users who might disseminate invalid or outdated information, the LBS information package (e.g., the set of points of interest) is proposed to be self-verifiable (i.e., be digitally signed by the server). In fact, this is the only change that MobiCrowd imposes on the LBS operation.

Our scheme leverages capabilities of contemporary smart phones: They can establish ad hoc and infrastructure connections (e.g., cellular base stations and Wi-Fi access points). We build a *mobile transparent proxy* in each device that protects the users' location-privacy. Our proxy, transparently located on-board the user's device and between the LBS client and the network, maintains a buffer with location context information. This buffer is checked for available data when the user submits a query. If the valid and up-to-date data is not available, our mobile proxy broadcasts the query (i.e., the type of required information) to other nearby devices. If and only if none of those neighbors can provide the requested information, is the LBS queried. We have implemented our scheme on the Nokia N800, N810 and N900 mobile devices, and demonstrated it with the Maemo Mapper (a geographical mapping software for points of interest) [30]. Note that our approach can be ported to the upcoming technologies that enable mobile devices to directly communicate to each other via (potentially more energy-efficient) Wi-Fi-based technologies [1], [2], [4] that aim at constructing a mobile social network between mobile users.

The rest of the paper is organized as follows. We survey the related work in Section II. In Section III, we state our model, the system assumption, and also the problem addressed in this paper. We present our scheme in Section IV, and then we develop an epidemic model of the MobiCrowd operation in Section V. We evaluate the effectiveness of MobiCrowd in Section VI, before we conclude the paper in Section VII.

II. Related Work

Techniques proposed to protect location privacy in LBSs can be classified based on how they distort the users' queries before they arrive at the LBS server. The queries can be *anonymized* (by removing users' identities) or *pseudonymized* (by replacing users' real names with temporal identifiers called pseudonyms), or they can be *obfuscated* (by generalizing or perturbing the spatiotemporal information associated to the queries). They can also be camouflaged by adding some *dummy queries*, or be completely eliminated and be *hidden* from the LBS [28]. Combinations of these methods have been employed in the existing (centralized or distributed) mechanisms. The interested reader is referred to [21], [28] for a more in-depth survey of the research on location privacy.

The mere anonymization of (especially the continuous) queries does not protect users' location privacy: the queries of a user are correlated in space and time, hence, the adversary can successfully link them by using target tracking algorithms [17] or identify the real names of the users [15], [20]. Changing user pseudonyms while the users are passing through pre-defined spots, called mix zones [7], makes it difficult to track the users along their trajectories. However, as users must remain silent inside the mix zones, so they cannot use the LBS, the size of the mix zones is kept small in order to let users benefit from the LBS. Thus, the unlinkability of users' queries is limited and the adversary's success is relatively high, even if the mix zones are optimally placed [12].

Perturbing the query's spatiotemporal information, in addition to anonymization by a a third party (central anonymity server), is proposed for obtaining a higher level of privacy [13], [24]. The main drawback is the reliance on a centralized third party that limits its practicality. The considerable degradation of the quality of service imposed by the obfuscation methods is another deterrent for such solutions. For example, in schemes such as [13], the queries sent to the anonymity server have to wait until enough anonymization can be achieved for a group of users (k-anonymity). Similarly in [8], the need to construct the cloaking regions and also to receive the responses from the server through other users can considerably degrade the service. Finally, most of the obfuscation-based techniques are based on k-anonymity, which has been shown inadequate to protect (location) privacy [31], [32].

Adding dummy queries to the user actual queries might help to confuse the adversary about the actual user location. But generating effective dummy queries that divert the adversary is a difficult task [9], as they need to look like actual queries over space and time. An optimum algorithm for generating dummy queries is an open problem.

In all the above-mentioned mechanisms, there is always a trade-off between users' privacy and the quality of service they experience. The tension is maximized when it comes to *hiding* queries from the LBS server. Hiding a query from the server minimizes the revealed user information, hence, maximizes her privacy with respect to that query. Simply put, it is more effective than the other three privacy protection methods, and it protects users against both presence and absence disclosure. This is what MobiCrowd provides: Hiding from the server while receiving the query responses from other peers.

Finally, there exist cryptographic approaches that redesign the LBS: the service operator does not learn much about the users' queries while it can still reply to their queries [14], or it can obtain imprecise information about user location [11]. The lack of incentives for LBS operators to change their business model and implement these solutions, and their relatively high computational overhead have made them impractical so far.

III. Problem Statement

A. System

We consider a network of location-aware wireless devices, capable of ad hoc device-to-device communication and of connecting to the wireless infrastructure (e.g., cellular and Wi-Fi networks). The users of such devices leverage on the infrastructure to reach the LBS servers. Users submit *localized search* queries, providing in principle their current location and the type of information (context, point of interest, etc) they are interested in. The server *replies* to them, providing the latest requested context information around the submitted location; e.g., on businesses, restaurants, gas stations, movie theaters, ongoing events, or current street traffic. The frequency at which users query the LBS varies depending on the type of requested information, the dynamics of information update in the LBS database, or the geographical region. We assume that the information the LBS provides is *self-verifiable*, i.e., users can verify that no entity (e.g., a compromised access point) changed the server reply content.

B. Adversary

LBS servers concentrate information about all user queries. Thus, an untrusted service provider could act as a "big brother," that is, it could monitor user whereabouts and activities over time. An honest but curious service provider could log the user interactions with the server and share them with other (untrusted) entities for monetary gain, e.g., for targeted advertisement. Moreover, the concentration of users' locations and other private information can attract criminals, who could break into the service provider network and steal this private information (with various malicious intentions). It is thus clear that location privacy is threatened by the LBS itself, which, at best, facilitates adversarial access to the user queries (and thus their locations and related private information). In such a setting, the adversary can be categorized as a *passive global long-term* observer, based on the terminology proposed in [28].

Inference attacks on the observed queries are classified into two tightly-related categories: *tracking* and *identification* attacks. Such attacks can lead to two types of location-privacy breaches: *presence* and *absence* disclosure. In other words, the adversary can learn that a user is at a given location, or that she is absent from certain locations, e.g., her home.

The more queries the adversary observes, the higher its location inference attack success will be. Less information about user locations makes it harder for the adversary to reconstruct their actual trajectories and to identify their real names. This is why protection mechanisms try to reduce the adversary's information. But, unfortunately, doing so reduces the quality of service for the user.

C. Design Objectives

Overall, we seek to design a practical and highly effective location-privacy preserving mechanism for LBSs. The nature of existing threats, outlined above, is the determining factor of our design objectives. The LBS business model itself can be at odds with the need to protect user privacy: LBS providers may actually need to profile users' activities, so that they can use such knowledge for various monetary purposes. As a result, the LBS operator may have no incentive to implement privacy-preserving mechanisms. In contrast, many users can be sensitive about their privacy. For this reason, our first design objective is to ***not** rely on architectural changes of the LBS*; any such changes (for example, using private information retrieval techniques [11]) would be impractical and highly unlikely to be adopted.

Moreover, relying on centralized trusted third parties (e.g., central anonymity servers) to provide privacy enhancing mechanisms can be as hard as having trusted LBS operators. In fact, as already mentioned, this would only shift the problem and such assumed trusted third parties would be new points of failure: once compromised, all users' information would be leaked to the adversary. This leads to our second design objective: *no reliance on any third party server to provide privacy protection*. In fact, we would like to *place the privacy protection* exactly where there is incentive and motivation, that is, *on the side of the users themselves*. We also want to achieve a high user privacy without sacrificing LBS quality of service by relying on users' collaboration.

IV. Our Scheme

Based on the stated design objectives, we propose a novel location-privacy preserving mechanism for LBSs. To take advantage of the high effectiveness of hiding user queries from the server, which minimizes the exposed information about the users' location to the server, we propose a mechanism in which a user can *hide in the mobile crowd* while using the service.

The rationale behind our scheme is that users who already have some location-specific information (originally given by the service provider) can pass it to other users who are seeking such information. They can do so in a wireless peer-to-peer manner, and in this way protect each other from privacy attacks that the adversary could perpetrate. Simply put, information about a location can "remain" around the location it relates to and change hands several times before it expires. Our proposed collaborative scheme enables many users to get such location-specific information from each other *without contacting the server*, hence minimizing the disclosure of their location information to the adversary.

A. Scheme Details

In order to better understand our model and solution, consider that the whole area covered by the roaming mobile users is divided into non-overlapping regions. Users can obtain context information associated to the region they find themselves in, e.g., obtain a list of businesses or services (and their latest status), or streets and intersections (and their traffic information). Users submit their queries when in place.

In this paper, without loss of generality, we focus on a single information type provided by the LBS (e.g., street traffic information, or oil prices in nearby gas stations, or a list of close-by restaurants). Clearly, users are interested in multiple types of location-based contextual information. The LBS server is responsible for compiling off-line the latest information for each region and for being ready to respond to the user query. The integrity and authenticity of the server responses is protected. This can be done in different ways; in our system, the user device verifies a digital signature of the

LBS on each reply using the LBS provider's public key. As a result, each piece of context information is self-verifiable: a compromised access point or mobile device cannot degrade the experience of users by altering replies or disseminating expired information.

Each piece of information associated with a given region has an expiration time (which is attached to the information and protected with the digital signature), after which the information is no longer valid. Every mobile device maintains a buffer in which location-specific information associated with regions is stored. This buffer keeps the replies the user obtains from the server or other peers. As long as a piece of information is not expired, it is kept in the buffer.

Each user with valid information about a region is termed *informed user*. Users interested in getting location-specific information about a region are called information *seekers* of that region. A seeker, essentially a user that does not have the sought information in her buffer, first broadcasts her query to her neighbors through the wireless ad hoc interface of the device. We term this a *local query*.

Any of the receivers of such a local query may respond to it, by what we term a *local reply*, as long as it has the information its peer seeks. However, an informed device will not necessarily respond to any received query: this will happen if the device is both *informed and willing to collaborate*. We design our system with this option for its users; the collaborative status may be set explicitly by the user or automatically recommended or set by the device. Simply put, having each user collaborate a limited number of times (a fraction of the times she receives a local query from her neighbors), or during a randomly selected fraction of time, balances the cost of helping other peers and caters to the needs of each user. In practice, this is equivalent to the case where only a fraction of users collaborate.

By obtaining a local reply, the seeker is now informed while, more importantly, her query has remained hidden from the service provider. No privacy-sensitive information has been exposed to the server and the user has obtained the sought service. Of course, in case there is no informed user around the seeker to assist her, she has no choice but to contact the server directly. In essence, a subset of users in every region have to contact the LBS to get the updated information, and the rest of the users benefit from the peer-to-peer collaboration. Intuitively, the higher the proportion of hidden user queries, the higher her location privacy will be.

V. The Epidemic Model

The performance of our system depends on various parameters, such as the frequency of contacts and the level of collaboration between users, the rate of query generation, etc. We now describe a model for MobiCrowd, with the help of which we can directly see the effect of various parameters on desired performance metrics. Observing the effect of the parameters helps when designing a system and testing "what-if" scenarios. For example, we can immediately see the level of collaboration required to achieve a desired privacy level or how the privacy level will change if the users make queries more frequently or less frequently.

We draw an analogy between our system and *epidemic phenomena*: location-context information spreads as an infection from one user to another, depending on the user state (seeking information, having valid information, etc.). For example, a seeker becomes "infected" when meeting an "infected" user, that is, a user with valid information.

We want a model that describes transitions between and keeps track of the various states a user is in at each point in time. However, the complexity of keeping track of each individual user state is prohibitive. Therefore, we make use of the *mean field approximation* [22], which focuses on the fraction of users in each state; these fractions are collectively called the *network state*. The approximation applies when the number of users is large and each individual interaction contributes a vanishingly small change to the network state. Also, the approximation requires a random contact pattern among users, rather than a spatially correlated pattern, and random contacts are not far from reality when users are clustered in the same area.

The mean field approximation tells us that the time evolution of the fraction of users in each state can be described with increasing accuracy, as the number of users grows, by a system of Ordinary Differential Equations (ODEs). By studying the system of ODEs, we find to what steady state(s) the network may converge to. Similar models have been used in epidemics [19], in worm propagation in wireless [33] networks, and also in research on forwarding/gossiping protocols [34].

To keep the presentation simple we focus on one type of context information, that is, we consider a single average information lifetime. No loss of generality results from this, because, to model a complete system with multiple types of information, we can merge multiple versions of this model, one for each type.

A. MobiCrowd: Model States and System of ODEs

As mentioned earlier, users move in an area partitioned into multiple regions. The state of context knowledge within a region intuitively corresponds to the disease status in an epidemic. In general, a user's knowledge state would be multi-dimensional, because it is different for each region. Hence, for each region we would have an associated epidemic model, with the same structure but different parameters. However, the state of knowledge about a region is unrelated to the knowledge about other regions, so different regions can be analyzed separately. We focus on a single region, with users entering and exiting it, and we describe the states and the dynamics of our epidemic model for that single region. The mobility of users with respect to a region is modeled using three parameters: β that is the average number of times a user have a proximity contact with other users at a time instant within a region, μ that is the average number of users who enter a region at a time instant, and λ that is the average number of users who leave a region at a time instant. The parameters of the epidemic model are listed in Table I.

Seeker*:* Users who are *inside the region* and are *interested in obtaining information* (i.e., have requested the information but not yet received it) are in the Seeker state. Once they have it, they move into the Informed state. Users can receive information from other Informed users in the region, or from the server, the ultimate source of information.

Informed*:* Users who *have information* about the region are in the Informed state. If they are *inside the region*, they (called Informed Insiders) accept to spread the information at each contact with a Seeker with probability ϕ. This is because the information spreading process imposes some communication cost on Informed users and, hence, they do not always collaborate. If they are *outside the region*, we assume they (called Informed Outsiders) do not spread the information (as nobody asks for it). The information that the Informed users have, whether they are inside or outside the region, expires with rate δ and the users become Removed.

Removed*:* Users who *do not have information* and are *not interested in obtaining information* are in the Removed state. We distinguish between Insider Removed and Outsider Removed users. An Insider Removed user becomes a Seeker if the user becomes interested in obtaining information about the region. We assume that outsiders have to enter the region to become interested.

We denote by $S(t)$, $I(t)$, $I^*(t)$, $R(t)$, and $R^*(t)$, respectively, the fraction of Seeker, Informed Insider, Informed Outsider, Removed Insider, and Removed Outsider users of a given region at time t. The *network state* $y(t)$ is the vector of these values. The time dependence will not be explicitly given in the rest of the paper. The system of equations that models the evolution of the network state is

$$S + I + I^* + R + R^* = 1 \tag{1a}$$

$$\frac{d}{dt}S = \gamma R - (\beta\phi I + \omega)S \tag{1b}$$

$$\frac{d}{dt}I = (\beta\phi I + \omega)S - \delta I + \mu I^* - \lambda I \tag{1c}$$

$$\frac{d}{dt}I^* = -\delta I^* + \lambda I - \mu I^* \tag{1d}$$

$$\frac{d}{dt}R = -\gamma R + \delta I + \mu R^* - \lambda R \tag{1e}$$

$$\frac{d}{dt}R^* = \delta I^* - \mu R^* + \lambda R \tag{1f}$$

$$0 \leq S, I, I^*, R, R^* \leq 1. \tag{1g}$$

We write this system succinctly as $\frac{d}{dt}y = F(y)$. We study the stationary regime of the system, i.e., the regime where, for $t \longrightarrow \infty$, the network state does not change with time. In particular, we look for equilibrium points of the system, i.e., network states at which $\frac{d}{dt}y = 0$. Setting $F(y) = 0$ and solving for y, we reach the following nonlinear system:

$$I = \frac{\omega S}{a - \beta\phi S} \tag{2a}$$

$$I^* = \frac{\lambda}{\mu + \delta} I \tag{2b}$$

$$R = \frac{\beta\phi I + \omega}{\gamma} S \tag{2c}$$

$$R^* = \frac{1}{\mu}\left(\frac{\lambda\delta}{\mu + \delta} I + \lambda \frac{\beta\phi I + \omega}{\gamma} S\right) \tag{2d}$$

$$\beta\phi S^2 - cS + a = 0, \tag{2e}$$

where

$$a = \delta\left(1 + \frac{\lambda}{\mu + \delta}\right) \tag{3}$$

$$c = a + \beta\phi + \omega\left(1 + \frac{\lambda}{\mu + \delta}(1 + \frac{\delta}{\mu}) + \frac{a}{\gamma}(1 + \frac{\lambda}{\mu})\right). \tag{4}$$

Having expressed all variables in terms of S, we need to solve the quadratic equation (2e) for S, keeping in mind that any solution has to satisfy $0 \leq S \leq 1$.

The value of S_0 can be found from the quadratic formula:

$$S_0 = \frac{1}{2\beta\phi}\left(c - \sqrt{c^2 - 4a\beta\phi}\right) \tag{5}$$

Then, we can substitute S_0 into (2a)-(2d) to find out the other values I_0, I_0^*, R_0, R_0^*.

So, we found the only admissible equilibrium point of the network. We now give a sufficient condition for this point to be locally asymptotically stable, that is, all system trajectories starting near enough to the equilibrium point will eventually converge to it without wandering too far away in the meantime. This condition

$S(t)$	Seeker users at time t
$I(t)$	insider Informed users at time t
$I^*(t)$	outsider Informed users at time t
$R(t)$	insider Removed users at time t
$R^*(t)$	outsider Removed users at time t
λ	rate of exiting the region per time unit
μ	rate of entering the region per time unit
β	contact rate per user per time unit
γ	avg request rate per user per time unit
$1/\omega$	avg waiting time before contacting the server
$1/\delta$	information avg lifetime
ϕ	avg collaboration probability

TABLE I
LIST OF THE SYMBOLS USED IN THE EPIDEMIC MODEL

is that the Jacobian matrix of the system, evaluated at the equilibrium point, has eigenvalues with strictly negative real parts. Note that, instead of using the differential equation for R^*, we substitute $R^* = 1 - S - I - I^* - R$ and compute the Jacobian of an equivalent system with only the 4 variables S, I, I^*, R. The Jacobian $J(S, I)$ is

$$\begin{pmatrix} -\beta\phi I - \omega & -\beta\phi S & 0 & \gamma \\ \beta\phi I + \omega & \beta\phi S - \delta - \lambda & \mu & 0 \\ 0 & \lambda & -\mu - \delta & 0 \\ -\mu & \delta - \mu & -\mu & -\gamma - \lambda - \mu \end{pmatrix} \tag{6}$$

which, as we see, is only a function of S and I. The eigenvalues of $J(S, I)$ evaluated at the equilibrium point can be found by solving the 4th order equation

$$|J(S_0, I_0) - xI_4| = 0 \tag{7}$$

for x, where I_4 is the 4×4 unit matrix. As we have mentioned, if all the solutions have a strictly negative real part, then the equilibrium point is locally asymptotically stable.

Moreover, if all the solutions have a strictly negative real part, the equilibrium point persists under small perturbations of the system. That is, if $v(y)$ is any smooth vector field on $\mathbb{R}^4$, then for sufficiently small ϵ the equation

$$\frac{d}{dt}y = F(y) + \epsilon v(y) \tag{8}$$

has an equilibrium point near the original one, and the equilibrium point of the perturbed system is also locally asymptotically stable.

In Section VI, we show that all the eigenvalues have strictly negative real part for the range of system parameters we consider; hence, the equilibrium point is stable, and it persists under small perturbations.

B. Baseline scenario: No collaboration

To be able to isolate the effect of collaboration, we study the case where there is no collaboration among users: A user who becomes interested checks her buffer, and if the content is not there, she immediately contacts the server. Thus, there are no S users in the model for this case:

$$I + I^* + R + R^* = 1 \tag{9a}$$

$$\frac{d}{dt}I = \gamma R + \mu I^* - (\lambda + \delta)I \tag{9b}$$

$$\frac{d}{dt}I^* = \lambda I - (\mu + \delta)I^* \tag{9c}$$

$$\frac{d}{dt}R = \delta I + \mu R^* - (\lambda + \gamma)R \tag{9d}$$

$$\frac{d}{dt}R^* = \delta I^* + \lambda R - \mu R^* \tag{9e}$$

$$0 \leq I, I^*, R, R^* \leq 1. \tag{9f}$$

We compute the equilibrium point of the system, and study its stability as before.

VI. EVALUATION

We evaluate the effectiveness of MobiCrowd in hiding user queries from the server, thus protecting their location privacy. First, we define a measure of the user privacy. Next, we simulate MobiCrowd on a dataset of realistic mobility traces and we compare the simulation results with the numerical results obtained from the epidemic model. Finally, we describe our implementation of MobiCrowd on the Nokia devices, and we present measurement results.

A. Privacy Gain

We quantify the privacy in a given region as the fraction of queries per time unit that are *not* observed by the server. This measure is inversely proportional to the adversary's success rate in performing inference attacks on the observed queries. This metric shows the reduction in the amount of information the adversary obtains from the users' queries *compared to* the case where users directly contact the server for each query.

1) MobiCrowd with no collaboration (relying on the buffer): In the case of no collaboration among users, which we use as a baseline scenario, the users can retrieve the information either from their buffer, or from the server. Only the I users have the information in their buffers, whereas the R users are forced to contact the server when they become interested. The I users ask queries at a total rate of γI, and the R users at a total rate of γR. Therefore, the privacy gain in this case is

$$PG_0 = I/(I + R) \tag{10}$$

where I and R are computed from (9).

2) MobiCrowd with collaboration: When the users collaborate with probability $\phi > 0$, queries can also be answered by peers, which happens at a total rate of $\beta\phi IS$. Queries are answered by the server at a total rate of ωS. The total rate of asked queries is, as before, $\gamma R + \gamma I$. Therefore, the privacy gain in this case is

$$PG_\phi = 1 - \frac{\omega S}{\gamma R + \gamma I} = \frac{\beta \phi I S + \gamma I}{\gamma R + \gamma I}. \qquad (11)$$

Observe that $\beta \phi I S$ is always smaller than γR (see Eq. (2c)), so the privacy is at most equal to 1, as it should be. The values of S, I and R are computed from (2).

B. Simulation Setup

In order to validate our model, we compare our numerical evaluations with simulation results. The location traces that we use belong to 509 randomly chosen mobile users (vehicles) from the epfl/mobility dataset at CRAWDAD [26]. We set the time unit of the simulation to 5 minutes and we consider the users' locations at integer multiples of the time unit, hence synchronizing all the traces. We consider a division of the Bay Area into 10×25 equal-size regions. Two nodes in a region are considered to be neighbors of each other if they are within 100m of each other (using WiFi). We run our simulation for 100 times on the mobility traces and compute the average of the results.

For each region, we compute λ, μ, and β from the data set. These values are plugged into the epidemic model in order to find the solutions of (2) and (9). We compute the privacy gain according to the simulation and the numerical analysis for $\phi = \{0.2, 1\}$, $1/\delta = \{1, 4, 7, ..., 28\}$, and $\gamma = \{0.1, 0.2, ..., 1\}$. The average waiting time before contacting the server $1/\omega$ is set to 1. For all combinations of these parameters, the eigenvalues of the Jacobians of (1) and (9) are negative, which indicates the stability of the equilibrium points of our epidemic model in these cases.

As it is not possible to plot the results for all the regions, we compute, as a representative example, the privacy gain in one region, located in downtown San Francisco. It has a higher concentration of points of interest, and 90 users are present in it on average. The rate μ of entering the region is 4.18 users per time unit, and the exiting rate λ is 4.22 users per time unit. The average contact rate β is 51.89 per user per time unit. In order to put the results from this region in perspective, we also measure the privacy of users across their entire trajectory, spanning multiple regions.

In the simulation we quantify the privacy gain using directly the definition (fraction of queries hidden from the LBS server), and for the numerical evaluation we use (10) and (11). We still use the notation PG_0 and PG_ϕ to refer to the location privacy gain of using MobiCrowd, without collaboration and with collaboration ϕ, respectively.

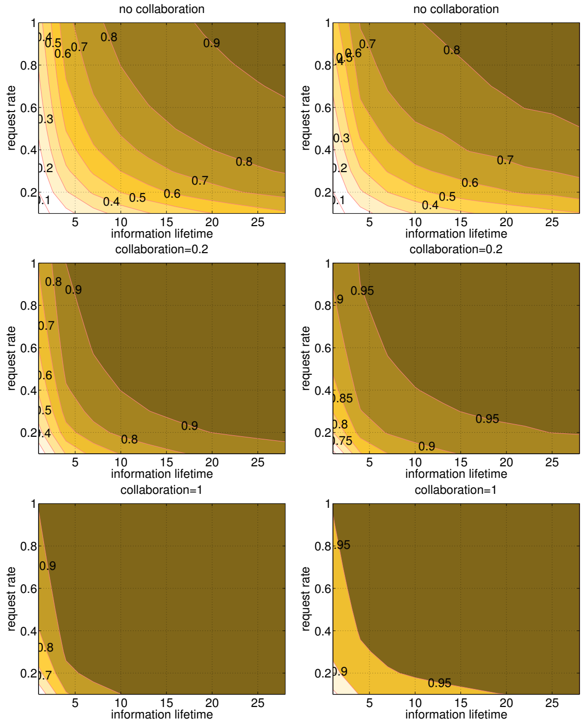

Fig. 1. User location privacy for the region under study (in downtown San Francisco). The first row illustrates privacy of users without collaboration, when they only rely on buffering information. The second and the third rows show user privacy using MobiCrowd, for different collaboration factors $\phi = 0.2$ and $\phi = 1$, respectively. The left column shows the numerical results whereas the right column shows the simulation results.

C. Results

Fig. 1 illustrates the users' location-privacy using MobiCrowd with and without collaboration (PG_ϕ and PG_0) in the studied region. The results of simulation and numerical evaluation are displayed side by side, in order to enable us to verify the validity of our epidemic model. The qualitative and also quantitative match between the simulation and the model enables us to rely on our epidemic model to evaluate users' location-privacy in a very computationally efficient way in complex scenarios dealing with large networks.

All the plots confirm a general pattern of privacy gain increase as the information lifetime or the request rate increases. With either kind of increase, users retrieve with higher probability non-expired information either from their own buffer or from their peers; hence, a higher fraction of their queries will be hidden from the LBS. Moreover, the privacy gain for long lifetimes and low request rate values (i.e., long intervals between requests) appears to be more or less the same as the privacy gain for short lifetimes and high request rate values (i.e., short intervals between requests), as indicated by the

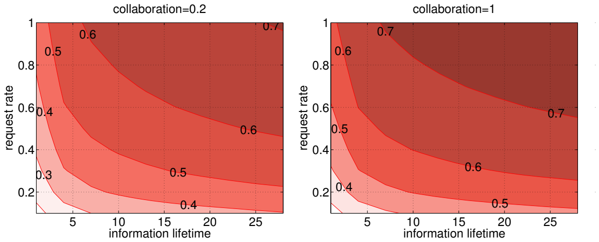

Fig. 2. Overall users' location-privacy using MobiCrowd across all regions, obtained by simulation.

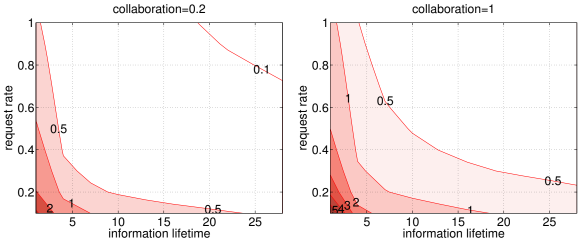

Fig. 3. Overall users' relative location-privacy gain of collaboration with respect to buffering in MobiCrowd across all regions, i.e., $(PG_\phi - PG_0)/PG_0$, obtained by simulation.

vaulted shape of the privacy gain contours. Also, adding collaboration to the buffering technique in MobiCrowd increases the fraction of hidden queries even for a collaboration factor of $\phi = 0.2$.

We observe these patterns in Fig. 1 which shows very high correlation between our epidemic model with the simulation of MobiCrowd on a realistic dataset. Even quantitatively, both sets of graphs match to a great extent. This proves the validity of our model in estimating users privacy gain even for the real scenarios where the contact rate between users changes over time.

Fig. 2 shows the simulation results for the users' privacy across their entire trajectory (over all the regions they visit) averaged over all the users. As we expect, increasing the collaboration probability increases user privacy, and the dependence on the information lifetime and the request rate is as we observed before in Fig. 1.

In Fig. 3, we see, again for the overall user privacy, the *relative* additional privacy gain we obtain by combining collaboration and buffering, compared to relying only on buffering. The relative added value of collaboration is computed as $(PG_\phi - PG_0)/PG_0$. So, for example, 0.5 on the plot means 50% increase in privacy gain.

We observe, first of all, that higher collaboration (going from $\phi = 0.2$ to $\phi = 1$) implies higher relative added value. What is more interesting, however, is that the relative privacy gain of collaboration increases as we go from the high-lifetime, high-request-rate part to the short-lifetime, small-request-rate part. In the former part, the effect of buffering dominates the privacy gain: The information does not expire quickly, so users retrieve it from their buffers, and so collaboration does not add much. Still, we observe relative gains of 10% even for low collaboration probability $\phi = 0.2$. In the latter part, however, the effect of collaboration dominates the achieved privacy, as buffering does not help much when the information lifetime is short: Increasing collaboration from 0.2 to 1 results in an increase of up to 500%. Summing up, buffering and collaboration complement each other in increasing user location-privacy.

The delay until receiving a response may be higher or lower with MobiCrowd: it depends on the implementation of the LBS, its workload at the time the query is sent, the available transmission capacity of the smartphones, and, above all, it depends on the state of the information in their buffer. In Section VI-D, we provide some information about the communication delay of MobiCrowd on Nokia devices.

D. Implementation

We implemented MobiCrowd on three different Nokia mobile devices (N800, N810, and N900). We built a *mobile privacy proxy* that runs in each device. The proxy does not require any modification of the supported applications and it is transparent to their operation. The prototype works with the Maemo Mapper LBS and MobiCrowd acts as a HTTP transparent proxy to which the client traffic is redirected. Note that knowing the format of the LBS queries and the data format of the server replies is enough to adapt MobiCrowd to new LBS applications (i.e., to parse the user queries and check whether the answer is in the buffer). Our implementation in Python (including the proxy module, ad-hoc networking module, and the server interface module) is 600 lines of code and the memory utilization does not exceed 3% of the total memory of the used devices.

We performed measurements to estimate the delay to obtain a peer response. The setting was a lab environment with 5 devices, 3 out of which were randomly chosen to collaborate each time. There were four POIs, and the size of the responses was 600 bytes. We average measurements over 100 queries. In our setting, the mobiles accessed the LBS server over a cellular link (e.g., GSM), and they communicated with other mobiles via the WiFi interface. The average delay was 0.17sec. We also note that cryptographic delays are (for a typical OpenSSL distribution) low: the weakest of the three devices, the N800, can verify more than 460 RSA signatures per second (1024 bit), or 130 signature verification per second (for 2048 bit modulus); this implies that digitally signed LBS response can be easily handled by the devices to protect against malicious peers.

A popular technique that enhances privacy against local eavesdroppers is to change the identifiers frequently. For example, in cellular networks the network operators are in charge of changing the TMSI when users move from one location area (a set of adjacent cells) to another. Thus, cellular networks make use of *network-issued pseudonyms* to protect the location-privacy of their users [5]. MobiCrowd-ready mobile devices can also mimic this defense (as has already been proposed for wireless networks, e.g., [18]). They can change their identifiers (e.g., the MAC addresses) as often as desired, even while in a single point-of-interest area. This would essentially root out any threat by any curious local observer. Even in the case of a stalker, it would not be possible to link the successive identifiers of a device to that device, as multiple users' identifiers will be mixed together. The only remaining option for the stalker is to maintain visual contact with the target user, but defending against this threat is clearly orthogonal to our problem.

Finally, our implementation allows the user to tune parameters (e.g., collaboration level).

VII. Conclusion

We propose a novel approach to enhance the privacy of LBS users, aiming against service providers who could extract information from their LBS queries and misuse it. We develop and evaluate MobiCrowd, a scheme that allows LBS users to reduce their exposure while they continue to receive the location context information they need. MobiCrowd achieves this by leveraging on peer collaboration: the user can get information from nearby users and can thus avoid getting exposed to the LBS server. Users, as opposed to the LBS server, have both the incentive and the capability to safeguard their privacy, thus they should be the ones responsible for it. Our analysis shows a significant improvement thanks to MobiCrowd, whose light-weight implementation we demonstrate in three mainstream portable devices.

Acknowledgment

The authors would like to thank Ehsan Kazemi for his valuable comments on the submitted manuscript. This work was supported (in part) by the National Competence Center in Research on Mobile Information and Communication Systems (NCCR-MICS), a center supported by the Swiss National Science Foundation under grant number 5005-67322.

References

[1] FlashlinQ: A Clean Slate Design for Ad Hoc Networks.

[2] NIC: Nokia Instant Community.

[3] Pleaserobme: http://www.pleaserobme.com.

[4] Wi-Fi Direct: http://www.wi-fi.org/wi-fi_direct.php.

[5] 3rd Generation Partnership Project. 3GPP GSM R99. In *Technical Specification Group Services and System Aspects*.

[6] R. Anderson and T. Moore. Information Security Economics–and Beyond. *Advances in Cryptology-CRYPTO 2007*.

[7] A. R. Beresford and F. Stajano. Mix zones: User privacy in location-aware services. In *PERCOMW*, 2004.

[8] C.-Y. Chow, M. F. Mokbel, and X. Liu. A peer-to-peer spatial cloaking algorithm for anonymous location-based service. In *GIS*, 2006.

[9] R. Chow and P. Golle. Faking contextual data for fun, profit, and privacy. In *WPES*, 2009.

[10] P. Eckersley. How unique is your web browser? In *PETS*, 2010.

[11] I. G. U. H. Femi Olumofin, Piotr K. Tysowski. Achieving efficient query privacy for location based services. In *PETS*, 2010.

[12] J. Freudiger, R. Shokri, and J.-P. Hubaux. On the optimal placement of mix zones. In *PETS*, 2009.

[13] B. Gedik and L. Liu. Location privacy in mobile systems: A personalized anonymization model. In *ICDCS*, 2005.

[14] G. Ghinita, P. Kalnis, A. Khoshgozaran, C. Shahabi, and K.-L. Tan. Private queries in location based services: anonymizers are not necessary. In *SIGMOD*, 2008.

[15] P. Golle and K. Partridge. On the anonymity of home/work location pairs. In *Pervasive*, 2009.

[16] M. Gruteser and D. Grunwald. Anonymous usage of location-based services through spatial and temporal cloaking. In *MobiSys*, 2003.

[17] B. Hoh and M. Gruteser. Protecting location privacy through path confusion. In *SECURECOMM*, 2005.

[18] T. Jiang, H. J. Wang, and Y.-C. Hu. Preserving location privacy in wireless lans. In *MobiSys*, 2007.

[19] W. O. Kermack and A. G. McKendrick. A contribution to the mathematical theory of epidemics. *Proc R Soc Lond A*, 1927.

[20] J. Krumm. Inference attacks on location tracks. In *Pervasive*, 2007.

[21] J. Krumm. A survey of computational location privacy. *Personal Ubiquitous Comput.*, 2009.

[22] T. G. Kurtz. *Approximation of population processes*. Society for Industrial and Applied Mathematics, 1981.

[23] C. Y. Ma, D. K. Yau, N. K. Yip, and N. S. Rao. Privacy vulnerability of published anonymous mobility traces. In *MobiCom*, 2010.

[24] J. Meyerowitz and R. Roy Choudhury. Hiding stars with fireworks: location privacy through camouflage. In *MobiCom*, 2009.

[25] M. F. Mokbel, C.-Y. Chow, and W. G. Aref. The new casper: query processing for location services without compromising privacy. In *VLDB*, 2006.

[26] M. Piorkowski, N. Sarafijanovic-Djukic, and M. Grossglauser. CRAWDAD data set epfl/mobility (v. 2009-02-24). Downloaded from http://crawdad.cs.dartmouth.edu/epfl/mobility.

[27] M. Piorkowski, N. Sarafijanovic-Djukic, and M. Grossglauser. A prsimonious model of mobile partitioned networks with clustering. In *COMSNETS*, 2009.

[28] R. Shokri, J. Freudiger, and J.-P. Hubaux. A unified framework for location privacy. In *HotPETS*, 2010.

[29] R. Shokri, J. Freudiger, M. Jadliwala, and J.-P. Hubaux. A distortion-based metric for location privacy. In *WPES*, 2009.

[30] R. Shokri, P. Papadimitratos, and J.-P. Hubaux. Mobicrowd: A collaborative location privacy preserving lbs mobile proxy. In *MobiSys - Demo Session*, 2010.

[31] R. Shokri, G. Theodorakopoulos, J.-Y. L. Boudec, and J.-P. Hubaux. Quantifying location privacy. In *SP*, 2011.

[32] R. Shokri, C. Troncoso, C. Diaz, J. Freudiger, and J.-P. Hubaux. Unraveling an old cloak: k-anonymity for location privacy. In *WPES*, 2010.

[33] G. Theodorakopoulos, J.-Y. Le Boudec, and J. S. Baras. Dynamic network security deployment under partial information. In *Allerton - Invited Paper*. 2008.

[34] X. Zhang, G. Neglia, J. Kurose, and D. Towsley. Performance modeling of epidemic routing. *Comput. Netw.*, 2007.