Private Location-Based Information Retrieval through User Collaboration

David Rebollo-Monedero [a,*], Jordi Forné [a], Agusti Solanas [b] Antoni Martínez-Ballesté [b]

[a] *Telematics Engineering Dept., Technical University of Catalonia*
C. Jordi Girona 1-3, E-08034 Barcelona, Spain
[b] *UNESCO Chair in Data Privacy, Dept. of Computer Engineering and Mathematics, Rovira i Virgili University*
Av. Països Catalans 26, E-43007 Tarragona, Spain

Abstract

Privacy and security are paramount in the proper deployment of location-based services (LBSs). We present a novel protocol based on user collaboration to privately retrieve location-based information from an LBS provider. Our approach does not assume that users or providers can be completely trusted with regard to privacy, and does not rely on a trusted third party. In addition, user queries, containing accurate locations, remain unchanged, and the collaborative protocol does not impose any special requirements on the query-response function of the LBS. Our protocol is analyzed in terms of privacy, network traffic, and processing overhead. We define the concept of guaranteed privacy breach probability, and we show that our proposal provides exponential scalability in that probability, at the expense of a linear relative network cost.

Key words: Location-based services, private information retrieval, location privacy, trusted third parties, untrusted user collaboration

1. Introduction

The opening up of enormous business opportunities for location-based services (LBSs) is the result of recent advances in wireless communications and positioning technologies. 3G technology makes mobile wireless communications faster than ever, and highly accurate positioning devices based on GPS are widely accessible to the general public [(1)] . Thus, due to the massive use of these technologies, an unprecedented amount of data is fleetingly traveling through high-speed networks all over the world. Naturally, some of these data refer to private user information such as user location and preferences and should be carefully managed.

There is a plethora of applications that utilize user locations to provide information or services. The following are some examples:

- The so-called concierge services are good examples of LBSs that will gain popularity in the near future. Their main goal is to guide users to the nearest point of interest, e.g., hospitals, restaurants, bus stops, or monuments.
- Entertainment and advertising services can also benefit from using the location of mobile users. Gamers will move from their computer desks to the streets and virtual worlds will be merged with real ones [2]. In addition, the way people buy will change thanks to the appearance of interactive location-based advertisements that will guide users to special offers or the like [3].
- Last but not least, emergency assistance services are also relevant examples of LBSs. When emergencies must be faced, location information automatically sent by means of a simple phone call can prove extremely useful to reduce the response time.

The way users access LBSs is changing rapidly. The simplest form of interaction between a user and an LBS provider involves a direct message from the former to the latter including a query and the location to which the query refers. An example would be the query "Where is

* Corresponding author. Tel.: +34 93 401 7027.
Email addresses: david.rebollo@entel.upc.edu (David Rebollo-Monedero), jforne@entel.upc.edu (Jordi Forné), agusti.solanas@urv.cat (Agusti Solanas), antoni.martinez@urv.cat (Antoni Martínez-Ballesté).

[(1)]The Apple iPhone 3G, equipped with GPS technology, was bought by more than 1 million people in the first weekend after its launch in July of 2008 [1].

the nearest bank?", accompanied by the geographic coordinates of the user's current location.

Under the assumption that the communication system used allows the LBS provider to recognize the user ID, there exists a patent privacy risk. More precisely, when users send their current locations to a service provider, they are implicitly trusting it because they assume that it will manage their location data honestly and will refrain from any misuse, such as profiling users according to their locations, the contents of their queries and their activity. However, providers may not always be trusted and, therefore, more complex communication schemes to achieve the same goals without assuming mutual trust are needed.

1.1. *Contribution and Plan of the Article*

We present a novel protocol that enables users to privately retrieve location-based information from an untrusted provider, bearing partial resemblance with mix networks. Our protocol does not rely on trusted third parties (TTPs) but on the collaboration among multiple untrusted users. Instead of using well-known techniques such as common cloaking areas or sharing of perturbed bogus locations, our method mixes queries from many users and prevents providers from binding queries to users, thereby protecting their privacy.

The rest of the paper is organized as follows. Section 2 summarizes the state of the art in privacy methods for LBSs. Section 3 presents our protocol for private, TTP-free, location-based information retrieval through user collaboration. A theoretical analysis of this protocol in terms of privacy, network traffic and processing cost is developed in Section 4. Section 5 sketches a preliminary analysis of the feasibility of our assumptions, and discusses a number of related issues. Conclusions are drawn in Section 6.

2. State of the Art

This section reviews the literature most particularly relevant to our proposed protocol for collaborative privacy applied to LBSs. More precisely, we provide a brief overview of the state of the art on privacy in LBSs and on anonymity based on mix networks, along with an introductory background regarding attacks against the latter systems.

2.1. *Privacy in LBSs*

A first step towards user privacy from the legal standpoint is the publication of privacy policies [4] by service providers stating what they may or may not do with the information collected, enabling users to take legal action against providers if needed. Under the platform for privacy preferences project (P3P) [5], the World Wide Web Consortium (W3C) has defined a protocol to systematize the way web sites such as e-commerce sites declare their use of information from browsing users. The geographic location privacy charter Geopriv of the IETF has proposed their use in LBSs [6].

An intuitive solution that would preserve user privacy in terms of both queries and locations is the mediation of a TTP in the location-based information transaction as depicted in Fig. 1. The TTP may simply act as an *anonymizer*,

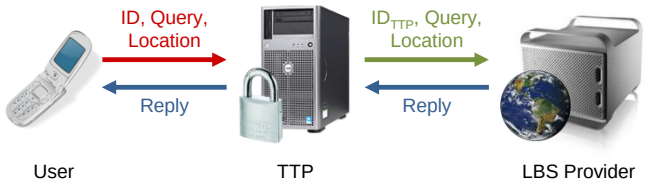


Fig. 1. Anonymous access to an LBS provider through a TTP.

in the sense that the provider cannot know the user ID, but merely the identity ID_{TTP} of the TTP itself inherent in the communication. Alternatively, the TTP may act as a *pseudonymizer* by supplying a pseudonym ID' to the provider, but only the TTP knows the correspondence between the pseudonym ID' and the actual user ID. A convenient twist to this approach is the use of *digital credentials* [7–9] granted by a trusted authority, namely digital content proving that a user has sufficient privileges to carry out a particular transaction without completely revealing their identity. The main advantage is that the TTP need not be online at the time of service access to allow users to access a service with a certain degree of anonymity.

Unfortunately, this approach does not prevent the LBS from inferring the real identity of a user by linking their location to, say, a public address directory, for instance by using *restricted space identification* (RSI) or *observation identification* (OI) attacks [10]. In addition, TTP-based solutions require that users shift their trust from the LBS provider to another party, possibly capable of collecting queries for diverse services, which unfortunately might facilitate user profiling via crossreferencing. Finally, traffic bottlenecks are a potential issue with TTP solutions, and so is any infrastructure requirement in certain ad hoc networks.

We shall see that the main TTP-free alternatives rely on perturbation of the location information, user collaboration and user-provider collaboration. The principle behind TTP-free perturbative methods for privacy in LBSs is represented in Fig. 2. Essentially, users may contact an

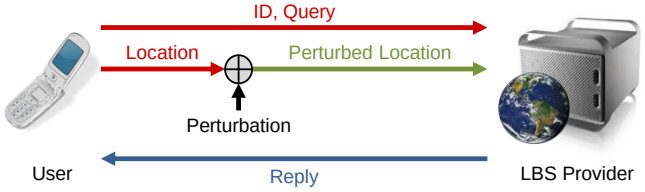


Fig. 2. Users may contact an untrusted LBS provider directly, perturbing their location information to help protect their privacy.

untrusted LBS provider directly, perturbing their location information in order to hinder providers in their efforts

to compromise user privacy in terms of location, although clearly not in terms of query contents and activity. This approach, sometimes referred to as obfuscation, presents the inherent trade-off between data utility and privacy common to any perturbative privacy method.

A wide variety of perturbation methods for LBSs has been proposed [11]. We cannot but briefly touch upon a few recent ones. In [12], locations and adjacency between them are modeled by means of the vertices and edges of a graph, assumed to be known by users and providers, rather than coordinates in a Cartesian plane or on a spherical surface. Users provide imprecise locations by sending sets of vertices containing the vertex representing the actual user location. Alternatively, [13] proposes sending circular areas of variable center and radius in lieu of actual coordinates. Finally, we sketch the idea behind [14]. First, users supply a perturbed location, which the LBS provider uses to compose replies sorted by decreasing proximity. The user may stop requesting replies when geometric considerations guarantee that the reply closest to the undisclosed exact location has already been supplied.

Fig. 3 is a conceptual depiction of TTP-free methods relying on the collaboration between multiple users. A proposal based on this collaborative principle considers groups of users that know each other's locations but trust each other, who essentially achieve anonymity by sending to the LBS provider a spatial cloaking region covering the entire group [15]. Recall that a specific piece of data on a particular group of individuals is said to satisfy the *k-anonymity* requirement (for some positive integer k) if the origin of any of its components cannot be ascertained, beyond a subgroup of at least k individuals [16, 17]. As many collaborative methods, the one just described guarantees k-anonymity regarding both query contents and location. Another effort towards k-anonymous LBSs, this time not assuming that collaborating users necessarily trust each other, is [18,19]. Fundamentally, k users add zero-mean random noise to their locations and share the result to compute the average, which constitutes a shared perturbed location sent to the LBS provider.

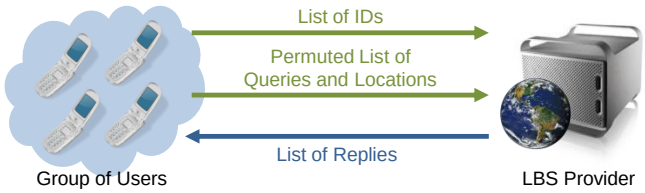


Fig. 3. Communication between a set of users and an untrusted provider without using a TTP.

A third class of TTP-free methods such as [20] builds upon cryptographic methods for *private information retrieval* (PIR) [21], which may be regarded as a form of untrusted collaboration between users and providers. Recall that PIR enables a user to privately retrieve the contents of a database, indexed by a memory address sent by the user, in the sense that it is not feasible for the database provider to ascertain which of the entries was retrieved [21]. Unfortunately, PIR methods require the provider's cooperation in the privacy protocol, are limited to a certain extent to query-response functions in the form of a finite lookup table of precomputed answers, and are burdened with a significant computational overhead.

Not surprisingly, a number of proposals for privacy in LBSs combine several of the elements appearing in all of the solutions above. Hybrid solutions more relevant to this work build upon the idea of *location anonymizers*, that is, TTPs implementing location perturbative methods [22], with the aim of hindering RSI and OI attacks, in addition to hiding the identity of the user. Many of them are based on the k-anonymity and cloaking privacy criteria [10, 18, 23–26].

Finally, we would like to briefly comment on some recent, rather sophisticated distributed protocols for privacy in LBSs, which also combine some of the ideas presented. While they do not assume that users necessarily trust each other, they do require certain trust relationships between architectural entities, or provide cloaking regions rather than exact positions to the LBS provider. The first example [27] envisions an architecture where two entities have access to user identities and location information separately. Accordingly, it is assumed that no information that could compromise location anonymity is exchanged between those two entities, for instance through auditing of a trusted mediator. An even more recent example [28] enables users to achieve location k-anonymity via a distributed, homomorphic cryptographic protocol. It is assumed that the location of each user is known by one server, and that servers do not collude with each other or with users. All the servers and each user jointly determine whether an area satisfies k-anonymity without the user revealing her precise location information to the other servers.

2.2. *Anonymization Based on Mix Networks*

Although the content of a message can be effectively protected by cryptographic techniques, this protection alone does not always guarantee anonymity. Namely, an attacker can observe the sender of a message and follow the message up to the receiver, thereby detecting the communication relation without any need to read the content of the transmitted message. *Traffic analysis* is the process of intercepting and examining messages in order to deduce information from patterns in communication. An example of protection mechanism, albeit rather obvious and potentially prohibitively inefficient, is traffic padding.

Chaum devised the idea of *mix networks* as a better solution to the problem of traffic analysis in a computer network. Fundamentally, a TTP called mix collects messages from a number of senders and forwards them to their intended receivers, possibly other mixes, rearranging them with the express purpose of hiding the correspondence between inputs and outputs. Messages sent to mixes are en-

crypted using public-key cryptography, in a layered fashion when several mixes are involved. Each mix strips off its own layer of encryption to reveal where to send the message next. Even if a single TTP performs both the role of user and mix, no generality is lost in our discussion by making a distinction.

There are several anonymous communication proposals based on the idea of mix networks. They can be roughly classified into high-latency and low-latency systems. The former introduce significant delay to attain a high degree of anonymity against traffic analysis [29, 30]. Naturally, their main drawback is that they are hardly applicable to real-time interactive tasks such as web browsing or online chat.

With the aim of overcoming this limitation, low-latency systems appeared. To attain a higher degree of anonymity, rather than increasing latency disproportionately, these systems simply benefit from the networking of a combination of several mixes frequently accessed by a significantly large population of users. Quoting [31], "All mix systems require that messages to be anonymized should be relayed through a sequence of trusted intermediate nodes". Some of these systems are based on peer-to-peer communications [32, 33], under the assumption of a very large interconnected population of trusted users who know how to reach each other, and who also act as mixes. The most popular approaches are *onion routing* [34, 35] and its second-generation version TOR [36]. Onion routing uses a single data structure encrypted in a layered fashion to build an anonymous circuit. Alternatively, TOR uses an incremental path-building design, where a client who wishes to communicate with a server negotiates session keys with each successive hop in the circuit. There exists a variety of mechanisms for building paths of mixes through which user messages will be forwarded [37], occasionally based on predetermined cascades of mixes. A large body of research, recently surveyed in [38], is concerned with extending and refining mix-based protocols.

Later on, in Section 3.4, once our own proposal will have been described in detail, we shall stress the differences between our protocol and mix networks.

2.3. *Attacks Against Mix Networks*

Since their introduction in the 80's [39], mix networks have been the subject of extensive studies concerned with their vulnerabilities, limitations and attacks. In the following, we cannot but roughly sketch some the main types of attacks studied. For a particularly comprehensive, up-to-date survey of attacks and proposals, please refer to [38].

Intersection attacks gain information about a targeted user through repeated observations of anonymity sets to which the targeted user belongs, that is, sets of possible identities for a given user determined on the basis of any observations. Since the intersection of two different anonymity sets is likely to be smaller than either of the anonymity sets, due to the assumed regularity in behavior, different intersections of anonymity sets could be used to gain information about the targeted user. Specifically, such intersection attacks may be based on replay, manipulation and blocking attacks [37].

Disclosure attacks, originally proposed in [40], are a formalized variation of intersection attacks. Assume a targeted user commonly communicates with m recipients. In a first, learning phase, the attacker waits until he observes m mutually disjoint sets of possible recipients for a targeted user, $R_1, \ldots, R_m$. In a second, excluding phase, further recipient sets inferred from new observations are intersected with $R_1, \ldots, R_m$ to refine them until all m recipients are singled out.

Statistical disclosure attacks [31, 41] are a statistical sophistication of the above attacks. The underlying principle may be expressed intuitively by saying that under reasonable assumptions, the uncertainty regarding the identity of the actual recipients of messages by a targeted user may be probabilistically reduced to an arbitrary degree, given a sufficiently large number of observations. Very recently, Bayesian probabilistic inference supported by Markov chain Monte Carlo numerical methods has been applied to deanonymize persistent communications [42].

A brief, preliminary discussion on the connection between these attacks against mix networks and our protocol may be found in Section 5.4.

3. A Collaborative Protocol for Privacy in LBSs

In this section we present a collaborative protocol that enables a number of users to interact with an LBS in a way that protects the privacy of their queries and replies. This is achieved in spite of two assumptions. First, it is assumed that neither the LBS nor other cooperating users can be completely trusted regarding the disclosure of a user's private information. Secondly, both the queries and the replies contain accurate information that may not be perturbed.

Section 3.1 makes our assumptions more precise. Performance criteria of interest are listed in Section 3.2. The privacy protocol proposed is described in Section 3.3, which relies on the existence of a cooperative structure of users. We stress the differences with respect to mix networks in Section 3.4. Section 4 analyzes our solution, mainly in terms of privacy risk and communication overhead. Later, Section 5 will sketch a preliminary analysis of the feasibility of our assumptions. In particular, security considerations concerning the creation and maintenance of such structure, including denial-of-service attacks against it, and the proper operation of the protocol, are discussed in Section 5.1. User unreliability and Sybil attacks are discussed in Sections 5.2 and 5.3. A preliminary analysis on the applicability of attacks intended against mix networks and traffic analysis is provided in Section 5.4.

3.1. *Hypotheses*

In the following, we describe the assumptions on which our collaborative privacy protocol has been built. We would like to remark that some are chosen intentionally to discard scenarios where privacy solutions have already been extensively studied. In particular, we rule out scenarios where the solution is as conceptually straightforward as introducing some form of TTP-based anonymization. The ulterior motivation is to provide a feasible solution to privacy for LBSs in ad hoc networks, networks where requirements regarding an underlying infrastructure must be kept to a minimum.

From the point of view of privacy and functional requirements, we shall assume the following:

- Users are allowed to cooperate but no party can be completely trusted, thus no TTP is available.
- Queries submitted to the LBS must be kept private and accurate, thus they may not be perturbed. In particular, noise may not be added to the original user location information to enforce privacy.
- The privacy protocol must be completely transparent to an arbitrary query-response function implemented by the LBS, without any significant collaboration of the provider. This prevents, for instance, the use of cryptographic mechanisms operating on the assumption of a limited response space, or a lookup table implementation of the query-response function, such as [20].

We elaborate on selfish, malicious and generally untrustworthy users in Section 5.2.

Regarding network communications, suppose that:

- Knowledge of the user ID is inherent in the communication system, and no form of anonymization is available, through a TTP or otherwise. IDs may neither be shared nor exchanged among users.
- All parties, that is, all users and the LBS, possess long-term IDs, and any two parties involved in a communication are capable of authenticating each other.
- Further, communication between any two parties is confidential and prevents traffic analysis.
- Messages exchanged between users may be encrypted for the LBS to further strengthen confidentiality. In practice, this would require that the LBS provider participate in any collusion against a user's privacy.

Although the above hypotheses essentially preclude Sybil attacks, we elaborate on this matter in Section 5.3. Note that our assumption regarding the existence of long-term IDs is not very restrictive. Otherwise, the concepts of privacy and anonymity would be hardly meaningful. In cases where IDs may be legitimately shared, the term user may simply be redefined as a disjoint user group, according to how IDs are shared, and the term privacy as group privacy. Section 5.4 elaborates on the use of cryptographic techniques to provide confidentiality and prevent traffic analysis, and briefly discusses the connection with known attacks against mix networks. Essentially, the hypotheses concerning secure network communication may be satisfied by the existence of a public-key infrastructure (PKI) [43], not necessarily online. The very last requirement, in particular, could be fulfilled by encrypting messages with the public key of the LBS provider.

Finally, we shall assume the existence of a secure mechanism by means of which users may organize themselves and adhere to a privacy protocol involving certain message exchanges. Particularly, we shall assume that there is a way to create and efficiently maintain collaborative structures, which is robust against denial of service attacks. Being not central to the focus of this work, but essential to its feasibility, the discussion of this assumption is postponed until Section 5.1.

3.2. *Performance Criteria*

Before describing our privacy protocol, we would like to state that we are interested in two types of performance criteria, namely privacy and network-related cost, which will be analyzed in Section 4. The main object of the protocol is to preserve user privacy. By privacy, we strictly mean anonymity of the query and the response contents, including location information, as a means to prevent any malicious party from profiling users according to their locations, the contents of their queries and their activity. Concordantly, we shall first consider the probability of events that compromise a user's privacy by means of collusion of malicious users and the LBS provider. Secondly, we shall assess the overhead cost in regard to network connections, traffic, LBS processing and latency.

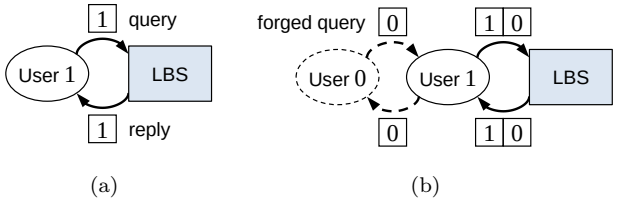
Additional performance criteria are resilience against denial of service attacks and costs related to the creation and maintenance of a collaborative structure enforcing the privacy protocol. Although these aspects are not analyzed in detail in Section 4, the discussion in Section 5 argues in favor of the feasibility of the protocol presented in this work.

3.3. *Description of the Protocol Proposed*

We now proceed to describe the collaborative protocol for privacy in LBS proposed in this work, assuming the hypotheses of Section 3.1. Before tackling the most general setting, special cases of gradually increasing complexity are discussed.

3.3.1. *Query Forgery for Private Information Retrieval*

Consider first the simplest case when a single user must access an LBS, depicted in Fig. 4(a), under the assumptions of Section 3.1. One way to ensure that the LBS provider is unable to completely ascertain the user's actual information interests is for the user to accompany his queries with forged ones, as illustrated in Fig. 4(b). In the figure, the authentic query is labeled with '1', and the forged one, with '0'. An interpretation that will be useful later consists in



(a) (b)

Fig. 4. A single user accessing the LBS provider. A forged query, accompanying the authentic one, provides a certain degree of privacy.

regarding the forged query as that generated by an imaginary user. The provider cannot discern which user submitted which query. Any policy regarding query forgery must take into consideration the trade-off among a number of factors, including not only resilience against statistical privacy attacks, but also traffic and processing overhead. Ultimately, the protocol described in this section will in fact accommodate any query forgery policy.

3.3.2. *Query Permutation on a Chain of Users*
We remarked that the single-user case of Section 3.3.1 could be regarded as the collaboration of two users, each submitting a query to the LBS provider, in such a way that the provider could not discern which query belonged to whom. The next setting we would like to consider, represented in Fig. 5, is an intuitively natural extension of the single-user case, in Fig. 4(b).

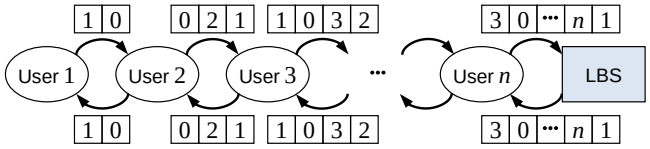


Fig. 5. Query permutation on a chain of users.

In this new setting, n users form a chain. User 1, exactly as in Section 3.3.1, accompanies authentic queries with forged ones in random order, but sends them to user 2 in lieu of the LBS provider. User 2 generates their own query, and then produces a random permutation of the forged query, user 1's query, and their own, labeled in the figure as '0','1' and '2', respectively. The randomly permuted list of queries is then sent to user 3, who in turn appends their query, and permutes the resulting list before sending it to user 4, and so on and so forth. The last user, user n, submits a random permutation of $n+1$ queries to the provider (one of them forged). The point is that neither the provider nor the intermediate users know for certain which authentic query was generated by whom.

To deliver the corresponding $n+1$ replies generated by the LBS, the process is reversed as shown in Fig. 5. Precisely, user j deletes their own reply from the list and inverts their permutations before sending it to user $j-1$. Note that user 2 sends two replies to user 1, one corresponding to the forged query. The overall mechanism is aimed at providing privacy regarding both queries and replies for all collaborating users, without assuming that the participants may be trusted. In principle, for each $1 < j < n$, user j's privacy can only be completely compromised if at least two users, namely $j-1$ and $j+1$, collude with each other to compare the lists of queries (replies) available to them, and pinpoint query (reply) j. We would like to remark that the confidentiality of the queries (but not the replies) sent from one user to another may be strengthened by an additional encryption reversible by the LBS provider only, e.g., encryption with the public key of the provider. In this way, two users and the provider would need to collude to reveal user j's query. Similarly, the integrity of the replies could be strengthened, provided that the provider agrees to sign them with its private key.

3.3.3. *Query Permutation on a Trellis of Users*
The most general setting proposed, up to small variations, is depicted in Fig. 6, where users form a trellis of m rows and n columns. At this point we would like to remark that

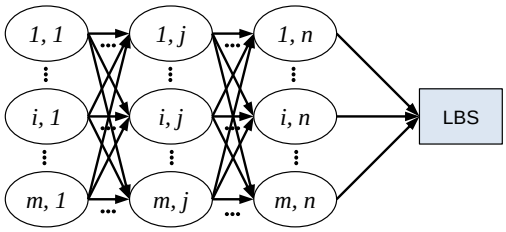


Fig. 6. Query permutation on a trellis of users.

an entirely analogous protocol to that described next could be carried out on a trellis with variable column size, with similar but not identical privacy properties. It will become clear that the scenario discussed in Section 3.3.2 is simply the special case when $m = 1$. In this new setting, users in the first column generate forged queries and send them along with their authentic queries to users in the second column. In general, as illustrated in Fig. 7, user (i, j) in row i and column j receives permuted queries from users in column $j-1$ when $j > 1$, or forges queries when $j = 1$. Next, the user adds their own query, permutes the resulting list, and finally, splits it and sends each part to different users in the following column $j+1$ if $j < n$, or to the LBS provider if $j = n$. The choices regarding the permutation of the list, its splitting, and the users the parts are sent to,

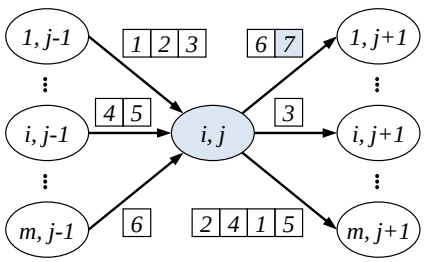


Fig. 7. User (i, j) adds their own query to those received from the previous column, permutes the resulting list, splits it and sends the parts to users in the next column.

are random. Replies are sent back reversing the permutation and splitting processes, similarly to the case $m = 1$ in Section 3.3.2.

The key difference with respect to the case with $m = 1$ is that more users need to collude to compromise a given user's privacy. More precisely, for $1 < j < n$, in order to guarantee that the privacy of user (i, j) be completely compromised regardless of how queries are split and transmitted, it seems that $2m$ users, namely all users from the previous column and the next one, must collude. However, $m + 1$ users are enough if we are satisfied with the coincidental configuration where all users in column $j-1$ collude with the user in column $j+1$ which happened to receive the query. A more detailed analysis is presented in Section 4.1.

Intuition suggests that the random query forwarding policy may be improved by enforcing maximum diffusion of queries from a user to the column of recipients. For example, if (i, j) has at least m queries, sending at least one to each of the users in $j + 1$ may have a positive impact in terms of probability of privacy breach. Such policy guarantees that users in $j + 1$ do not need to generate forged queries to protect their privacy against users in $j + 2$. On the other hand, users in the first column would still need to forge $m - 1$ queries each, to enforce this throughout the trellis.

3.3.4. *Variations of the Privacy Protocol*

There exists a number of variations and twists on the protocol described in Section 3.3, which may be of help when tailoring an implementation to fit the requirements of a particular application, with consequent variations in performance in terms of privacy and number of messages. Two examples follow.

Consider first each of the chains of n users along a row of the trellis in Section 3.3.3, which is also the case discussed in Section 3.3.2. The number of queries processed by the LBS provider for each row can be reduced to the minimum number n of authentic queries for any $n \geqslant 3$, as shown in Fig. 8. The only difference with respect to the previous

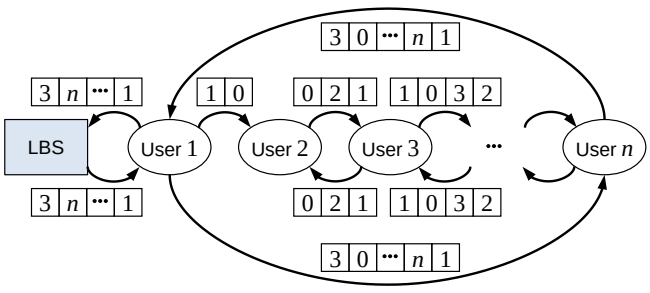


Fig. 8. Query permutation on a chain of users where the forged query is removed from the list sent to the LBS.

protocol is that user n now sends the permuted list of $n+1$ queries back to user 1 instead of submitting it to the LBS. Then, user 1 removes the forged query, sends the resulting list to the provider, and adds a forged reply to the received list, which is sent to user n.

As a second example of variation, for $1 < j < n$, user (i, j) in the trellis could split their own query into portions sent to all users in column $j + 1$. In this way, a single malicious user in the next column does not suffice for a complete privacy breach. These portions should be properly tagged in order for the LBS provider to recombine them. However, if all users in column $j + 1$ are malicious, they could keep track of the recombination tags to discard incomplete groups of query portions coming from (i, j), in order to compromise the user's privacy. At the other extreme, an alternative to reduce the number of messages would consist in sending the entire query list to a single, randomly chosen user from the next column, as depicted in Fig. 9.

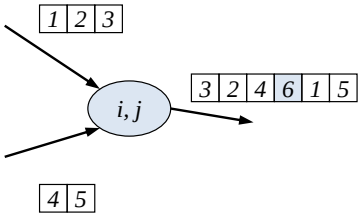


Fig. 9. Case when user (i, j) sends the entire list of queries to a single user in the next column.

3.4. *Differences with Respect to Mix Networks*

We briefly discussed mix networks in Section 2.2 as a TTP-based, multiparty solution to anonymize network traffic. We now proceed to stress the differences between our protocol for privacy in LBSs through user collaboration, and mix networks. First, the main motivation for our protocol is to minimize infrastructure requirements by excluding the involvement of TTPs such as anonymizers or mixes. Accordingly, as stated in Section 3.1, we do not assume that any of the parties involved in the query exchange, not even the users themselves, can be completely trusted. That is in fact one of the reasons for requiring query forgery. In other words, in our work, users are *not* assumed to act as mixes, in the sense that while they collect, rearrange and forward messages, are far from being conceivable as TTPs. Clearly, the above does not preclude the management of encryption keys from relying on TTPs as part of a PKI, although they would not need to be online during the query communication process. In any case, we do not impose the requirement for additional TTPs.

On the other hand, it is only fair to acknowledge that threat models for mix networks do consider the possibility of attackers "who can compromise some fraction of the onion routers" [36]. Further, the introduction of bogus queries as a means to protect privacy is by no means new, as the idea was already introduced in [39].

Secondly, our proposal and mix networks have different goals. Specifically, the goal of our protocol is to anonymize a location-based query completely, against the recipient itself, namely the LBS provider, whereas mix networks strug-

gle to prevent traffic analysis aimed at tracing the connection between senders and receivers.

Despite these differences, many of the attacks against mix networks might still apply against our protocol. This is further discussed in Section 5.4.

4. Privacy and Cost Analysis

In this section we analyze the trellis structure described in Section 3.3.3 in terms of two contrasting aspects. On the one hand, we consider the usefulness of this structure to enforce user privacy. On the other, we study the overhead cost in regard to network connections, traffic, LBS processing and latency.

4.1. *Privacy*

There exists a number of statistical quantities that may serve as an insightful measure of a user's privacy in the context of our work. In this section, we are inclined to focus on a simple quantity ensuring a high degree of mathematical tractability. Specifically, we shall compute the probability that the anonymity of the query of a given user is completely compromised by a group of colluding users. Of course, if mathematical tractability were not an issue, more complex privacy measures might have been proposed. For example, entropy measurements based on the probability that a query belongs to a user, within a list obtained after collusion. Or inspired by [44, 45], we could have alternatively defined privacy risk as the Kullback-Leibler divergence between the user's probability distribution of query values and the population's. A discussion of privacy measures is provided in [46].

4.1.1. *Assumptions*

Interested in a conceptual, preliminary analysis, we shall simply assume that users disclose their lists of queries and are willing to collude with other users to compromise a given user's privacy, with identical probability $1-t$, independently from each other, conditioned on the event that the LBS provider is willing to act maliciously as well. Loosely speaking, t is the probability that a user can be trusted, given that the LBS cannot. Conditioning on the event that the provider is malicious makes the computation identical regardless of whether queries are encrypted for the LBS, and yields slightly simpler expressions, omitting a constant factor, namely the probability that the provider acts maliciously. Accordingly, the conditional probability of exactly k users colluding with each other against another user's privacy, given that the provider maliciously cooperates, is $(1-t)^k$. More realistic scenarios could of course be better characterized by more complex probability models. For example, the risk of being caught might deter a user from working in collusion with too many others, making the corresponding probability even smaller than the power considered here. In addition, also for the sake of simplicity and despite its importance, our analysis neglects the risks derived from exploiting any statistical dependence between queries and replies, and any statistical dependence among numerous repetitions of the protocol proposed in Section 3.3.3. Confidentiality and traffic analysis are discussed in Section 5.4. Finally, our privacy analysis focuses only on queries, rather than replies, due to the similarity of the alternative analysis.

4.1.2. *Probability of Complete Privacy Breach*

Privacy is not completely compromised as long as a list of queries contains at least one query in addition to the user's. Provided that users in the first column of the trellis of Section 3.3.3 submit forged queries, any group of colluding users will be unable to ascertain authentic queries, at least without further statistical analysis, according to the assumptions stated above.

For $j > 1$, consider the case when user (i, j)'s query is known to a group of users colluding with each other and the LBS provider. The probability that this situation is guaranteed to happen regardless of how query lists are split and transmitted, requires collusion of the $2m$ users in columns $j-1$ and $j+1$, or merely the m users in column $j-1$ if $j = n$. According to our hypotheses, this probability of *guaranteed complete privacy breach* (GCPB) is $p_{\mathrm{GCPB}} = (1-t)^{2m}$ for $1 < j < n$, and $(1-t)^m$ for $j = n$. Let f denote the amount of forged queries generated by users in column 1. As long as $f > 0$, it is clear that $p_{\mathrm{GCPB}} = 0$ for $j = 1$. Observe that by definition p_{GCPB} is a probability conditioned on the event that the LBS provider acts maliciously, in cooperation with the group of colluding users. Consequently, the unconditional probability may be even lower. Table 1 evaluates $(1-t)^{2m}$ for several values of t and m, and illustrates what is perhaps the strongest point of the trellis structure proposed, namely an exponentially scalable probability of GCPB.

More likely is the case when user (i, j)'s privacy is completely compromised due to a coincidental configuration of malicious users and query list splitting, without the a priori guarantee. For $1 < j < n$, for instance, the query of interest will be disclosed if all m users in column $j-1$ and the single user in column j where the query is actually sent collude. We show in Appendix A that under the mild assumption of symmetry and random query forwarding, the probability of *coincidental complete privacy breach* (CCPB) is, for $1 < j \leqslant n$,

$$p_{\mathrm{CCPB}} = (1-t) \sum_{b_1=0}^{m} \cdots \sum_{b_{j-1}=0}^{m} \left(\prod_{k=1}^{j-1} \binom{m}{b_k} \right) \times t^{\sum_{k=1}^{j-1} b_k} (1-t)^{m(j-1) - \sum_{k=1}^{j-1} b_k} \times \left(1 - \frac{\prod_{k=2}^{j-1} b_k}{m^{j-1}} \right)^{b_1 (1+f)}$$

t	m	$1/p_{\mathrm{GCPB}} = 1/(1-t)^{2m}$
1/2	1	4
1/2	2	16
1/2	3	64
1/2	5	1 024
1/2	10	1 048 576
4/5	1	25
4/5	2	625
4/5	3	15 625
4/5	5	9 765 625
4/5	10	$\simeq 9.536 \cdot 10^{13}$
9/10	1	10^2
9/10	2	10^4
9/10	3	10^6
9/10	5	10^{10}
9/10	10	10^{20}

Table 1
Inverse probability of guaranteed complete privacy breach $1/p_{\mathrm{GCPB}}$ for $1 < j < n$, given that the LBS provider is malicious.

$$\times \prod_{k=2}^{j-1} \left(1 - \frac{\prod_{l=k+1}^{j-1} b_l}{m^{j-k}}\right)^{b_k}. \quad (1)$$

Of course, $p_{\mathrm{CCPB}} = 0$ for $j = 1$ as long as the number of forged queries $f > 0$. Implicit in the assumption of random query forwarding is the possibility that some intermediate users not receive any queries, which contributes to the probability of CCPB in the above formula. In a practical situation, since the (i, j) user sends $j+f$ queries on average, this can be easily avoided for $j \geqslant m - f$ by requiring that at least a query be forwarded to each user, by making f large enough, or by allowing intermediate users to forge queries as well. Concordantly, the formula is best interpreted as an upper bound on the probability of CCPB.

The appendix also proves the symbolically simpler bounds

$$(1-t)^{m+1} \leqslant p_{\mathrm{CCPB}} \leqslant (1-t)(1-t/m)^m. \quad (2)$$

Both the upper and lower bound are tight for $m = 1$, but not in general. To see the the lower bound is not tight, for instance, note that the only malicious users required in columns $1, \ldots, j-1$ are those observing queries in the list received by the single malicious user in column $j+1$, such that all queries but (i, j)'s can be discarded. The upper bound can be loosened on account of the fact that

$$(1-t)(1-t/m)^m \leqslant (1-t)e^{-t},$$

inequality asymptotically tight as m goes to infinity.

The derivation in the appendix concludes with the approximation for $t \simeq 1$

$$p_{\mathrm{CCPB}} = (1-t)\left(1 - \frac{1}{m}\right)^{m(j+f-1)} + o(t-1). \quad (3)$$

This approximation may be interpreted as the probability that user $(1, j+1)$ is malicious, and that although no user acts maliciously in columns $1, \ldots, j-1$, unfortunately no queries are received by user (i, j). If the query forwarding policy were not random, but instead guaranteed that at least one query reaches each intermediate user, the first-order approximation shown would be trivially zero.

Fig. 10 depicts the probability of CCPB according to (1), plots the bounds (2), the approximation (3), and includes the much lower probability of GCPB $(1-t)^{2m}$ as reference. The plots suggest that larger m and f help reduce p_{CCPB}, in keeping with the exact formula (1). The fact that the approximation (3) becomes $(1-t)\,e^{-(j+f-1)}$ in the limit of infinite m means that, for trustworthy users, p_{CCPB} decreases exponentially with j and f, but approaches a saturation level for large m. This phenomenon is supported by the curves depicted for $t = 9/10$ in Fig. 10.

The intraquery splitting alternative commented on in Section 3.3.3 may be an additional degree of freedom in the protocol to alter the probability of CCPB, but the probability of GCPB will remain equal to $(1-t)^{2m}$. While both probabilities can always be reduced by generating additional forged queries at intermediate users, this comes at the cost of network traffic, and LBS processing time.

4.2. *Cost*

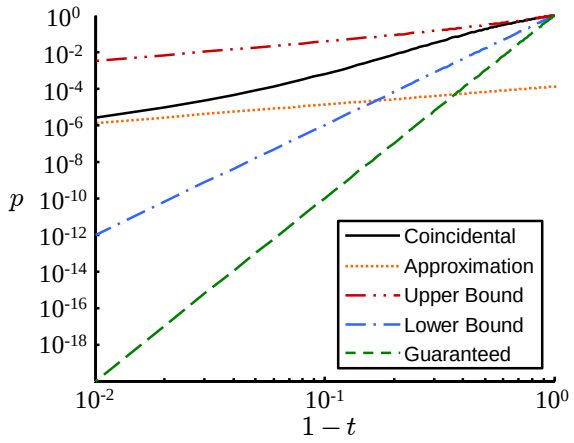
We now turn to considerations of cost, briefly analyzing the overhead incurred by the privacy protocol on a trellis of m rows and n columns, in regard to network connections, traffic, LBS processing and latency. The analysis is restricted to queries. The corresponding analysis for responses is entirely analogous. For simplicity, costs regarding structure creation and maintenance, for example using the procedures discussed in Section 5.1, are not taken into account, although we acknowledge they may be noticeable, particularly in mobile environments with long backhaul links.

Results are summarized in Table 2. It is important to

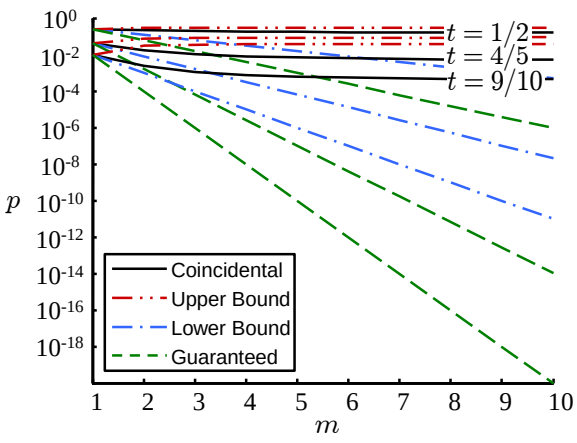
Max. # of messages transmitted	$m^2(n-1)+m$	$O(m^2 n)$
Relative overhead (w.r.t. $m n$)	$\frac{m(n-1)+1}{n}$	$O(m)$
# of queries transmitted	$m n \left(\frac{n+1}{2} + f\right)$	$O(m n^2)$
Relative overhead (w.r.t. $m n$)	$\frac{n+1}{2} + f$	$O(n)$
Min. # of avg. queries per message	$\frac{n\left(\frac{n+1}{2}+f\right)}{m(n-1)+1}$	$O\left(\frac{n}{m}\right)$
# of queries processed by LBS	$m(n+f)$	$O(m n)$
Relative overhead (w.r.t. $m n$)	$(1 + f/n)$	$O(1)$
Latency	$\simeq \frac{m n}{\lambda}$	$O(m n)$

Table 2
Summary of cost analysis, assuming f does not scale with m or n.

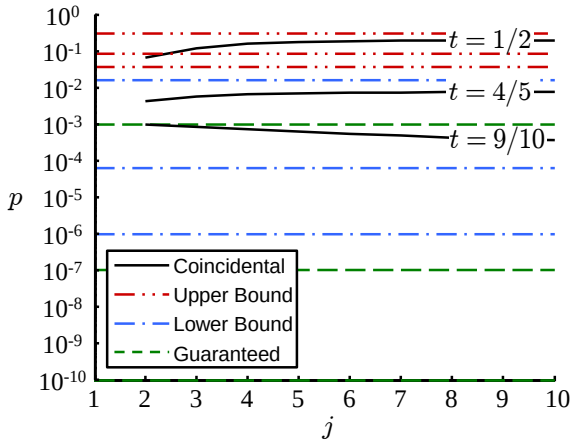
note that the results in Landau asymptotic notation implicitly assume that f does not scale with m or n. Under a random query forwarding policy, this entails the possibility that intermediate users may not receive queries from previous columns. This rules out the case when $f = m - 1$, mentioned in Section 4.1.2, in which all users are guaran-



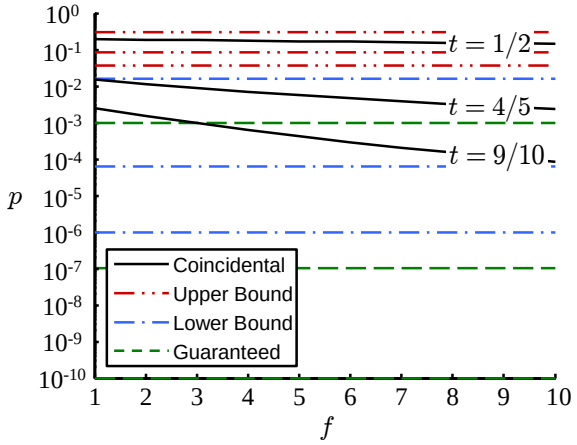
(a) $m = 5, j = 5, f = 4$



(b) $m = 1, \ldots, 10, j = 5, f = 4$



(c) $m = 5, j = 2, \ldots, 10, f = 4$



(d) $m = 5, j = 5, f = 1, \ldots, 10$

Fig. 10. Probability of CCPB (1), together with the upper and lower bounds (2), the approximation (3) for $t \simeq 1$, and probability of GCPB.

teed to receive queries from previous columns, as long as queries are diffused properly.

First, we claim that the number of messages containing queries in the trellis, equal to the number of connections, is bounded from above as follows:

$$N_{\text{messages}} \leqslant m^2\,(n-1) + m = O(m^2\,n).$$

Indeed, each of the m users in each of the first $n-1$ columns may communicate with up to m users in the next column. This represents $m^2\,(n-1)$ connections, to which we need to add the m connections with the LBS established by the m users in the last column. If privacy were not an issue, the $m\,n$ users in the trellis could directly send their queries to the LBS provider, which would require only $m\,n$ query messages. The relative overhead of the (maximum) number of query messages N_{messages} with respect to the ideal minimum $m\,n$, attainable with perfect trust, is the ratio $\frac{m\,(n-1)+1}{n}$, asymptotically equal to m as n goes to infinity.

We show now that the total traffic measured in number of queries transmitted through the trellis and to the LBS provider is

$$N_{\text{queries}} = m\,n\left(\frac{n+1}{2} + f\right) = O(m\,n^2),$$

where f is the number of forged queries generated by each user in the first column. The claim follows from the fact that a total of $m(j+f)$ queries are transmitted from users in column $j = 1, \ldots, n$ to either users in column $j+1$ if $j < n$, or to the provider otherwise. The grand total of queries is then

$$\sum_{j=1}^{n} m(j+f) = m\left(\sum_{j=1}^{n} j + n\,f\right),$$

equal to the expression claimed. If the LBS were completely trusted, then a single query could be generated by each of the $m\,n$ users and transmitted directly. Hence, the relative traffic overhead is $\frac{n+1}{2} + f$, which approaches the linear function $n/2$ as n goes to infinity.

We may draw two corollaries from the previous analysis. On the one hand, we already mentioned that the number of queries received by the LBS provider is $m(n+f) = O(m\,n)$, which can be regarded as the LBS processing cost, with relative overhead $1 + f/n$, vanishing in the limit of

large n [(2)] . On the other hand, the quotient

$$\frac{N_{\text{queries}}}{N_{\text{messages}}} \geqslant \frac{n\left(\frac{n+1}{2}+f\right)}{m(n-1)+1} = O\left(\frac{n}{m}\right)$$

gives the average query message length, measured in number of queries, provided that the maximum number of messages N_{messages} is attained, or more generally, a lower bound. This quotient is approximately $\frac{n}{2m}$ for m and n large.

One of the strengths of the collaborative protocol proposed becomes apparent in light of the above cost calculations and the privacy analysis of Section 4.1.2. Specifically, we benefit from an exponentially decreasing probability of GCPB, at the expense of linearly increasing relative overheads.

Finally, one of the fundamental performance criteria of the protocol is clearly the latency inherent to the fact that users must wait for others to cooperate before sending their queries, similarly to mix networks. Intuitively speaking, in the special case when users only participate in the protocol when they have a query available, a latency constraint in turn imposes an upper bound on the average number of participants. In light of the privacy analysis of Section 4.1, this suggests the existence of a trade-off between privacy and latency. Of course, a more forgiving trade-off could be attained in the case when users participate regardless of whether they have a query available.

The following back-of-the-envelope analysis will help us quantify this statement. Suppose that the entire population of users generates (authentic) queries according to a Poisson process at a rate of λ queries per time unit. Let L denote the maximum amount of time a user is willing to wait from the instant it starts to form a new trellis, until the trellis is completely filled with users willing to query the LBS provider. Again, assume users only participate in the protocol when they have a query available. Then, the number of *additional* users available to join the trellis during this period is a Poisson random variable with expectation λL. Consequently, the expectation of the number $m\,n$ of users in this new trellis is at most $1 + \lambda L$. For the sake of argument, assume that the population is large and sufficiently active, disregard the possibility of repeated queries from the same user within a short latency period, and that consequently, roughly, $m\,n \simeq \lambda L$. Under this approximation, for example, a total population of 100 000 users, each of them producing 1 query per hour but willing to tolerate a latency of only 1 second, may be able to participate in trellises of the order of up to

$$m\,n \simeq \frac{100\,000}{1\,\text{hour}}\,1\,\text{second} = \frac{100\,000}{60\times 60} \simeq 28$$

members with similar behavior.

[(2)]If f does however scale with m, as in the variation $f = m - 1$ described in Section 4.1, then the number of queries processed by the LBS is $O(m(m+n))$, and the relative overhead $O\left(1+\frac{m}{n}\right)$.

5. Discussion

We proceed to discuss in greater detail the assumptions of Section 3.1. Even though the focus of this paper is to present and analyze a collaborative privacy protocol, we would like to provide at least a brief analysis of feasibility, with emphasis on the creation and maintenance of a collaborative structure supporting our protocol. Concordantly, Section 5.1 suggests that a reasonably secure underlying structure is feasible, at least in certain scenarios. Next, we elaborate on the rest of hypotheses of Section 3.1. Our discussion occasionally relies on the existence of a PKI. The feasibility of PKIs in mobile ad hoc networks (MANETs), one of our scenarios of interest, is tackled in [43].

5.1. *Creation and Maintenance of a Collaborative Structure*

We consider now the creation and maintenance of the structure assumed at the end of Section 3.1, which serves as basis for the privacy protocol described in Section 3.3, particularly in the form of the trellis of Section 3.3.3. A simple protocol is presented first, merely as an argument in favor of the feasibility of the privacy protocol, under small-scale scenarios. We emphasize at this point that the efficient, robust, secure creation and maintenance of a large-scale, long-lived collaborative structure is by no means a trivial problem, but an interesting open challenge. Indeed, the feasible sizes of such collaborative structure, and its precise suitability to static and dynamic networks, may not be trivial matters, and they are ultimately subject to the specific requirements and acceptable trade-offs of the target application, including at least considerations of privacy, traffic overhead and latency. Fortunately, the privacy analysis of Section 4.1 suggests that even a small number of participants may suffice, as the probability of GCPB scales exponentially, fact illustrated in Table 1.

Many of the hypotheses for our privacy protocol in Section 3.1, discussed in greater detail in the following subsections, are just as relevant to our discussion on the creation and maintenance of the structure. For example, we assume Sybil attacks are not an issue in the creation and maintenance of the structure, according to the authentication hypotheses of Section 3.1, on which we elaborate in Section 5.3.

5.1.1. *Simple Scenario*

Focusing on feasibility rather than efficiency, we outline a simple, small-scale version of a protocol for creating a short-lived collaborative structure. Basically, we propose user grouping via broadcast or multicast communication. In addition, we define an ordering in the trellis structure indirectly determined by the user IDs of its members, rather than an ordering dictated by a cluster coordinator. The purpose of this ordering is to make it difficult to collude against a user's privacy through strategic positioning in the trellis.

More specifically, recall that we assumed in Section 3.1 that, essentially, user IDs where not interchangeable and user communication was authenticated. Consistently, consider a group of N users with authenticated identities $\mathrm{ID}_1, \ldots, \mathrm{ID}_N$. Let h be a common hash function, known by all users, with the usual cryptographic properties, including unfeasible computation or modification of input given a fixed output. In the following, we shall refer to the bitstring alphabetical ordering of $h(\mathrm{ID}_1), \ldots, h(\mathrm{ID}_N)$ as the *ID hash ordering* of the N users. In the trellis setting of Section 3.3.3, assume that vacant positions, either during a creation or a maintenance phase, are filled rowwise, i.e., in the order $(1,1), (1,2), \ldots, (1,n), (2,1), \ldots, (m,n)$, in accordance with the ID hash ordering of its members. The use of $h(\mathrm{ID})$ in lieu of the ID value should hinder colluding attackers in their efforts to compromise a user's privacy by positioning themselves in sensitive locations of the trellis, particularly the previous and next columns.

As for grouping via broadcast, we assume that user clusters will be formed by a small number of users capable of intercommunication. The creation of such clusters may be accomplished by means of the following simple protocol:

1. When user u wants to create a cluster, it broadcasts a request message $\mathrm{JoinREQ}(u)$.
2. Any user v in the communication range of u, with a query ready to be sent, wishing to take part in the collaborative structure, broadcasts the acknowledgment $\mathrm{JoinACK}(u, v)$.

Users that receive both the $\mathrm{JoinREQ}(u)$ request and the $\mathrm{JoinACK}(u, v)$ acknowledgments are included in the cluster. In this way, all users in the group know the identities of its participants, together with the specific positioning in the trellis structure, determined by the ID hash ordering described above.

5.1.2. *More General Considerations*

Having presented a simple protocol for the creation of a collaborative structure as a feasibility argument, we now comment on some of the issues involved in tackling a more general version of the problem, more efficiently, thereby widening its range of applications. For example, the simple protocol of Section 5.1.1 only considers the creation but not the maintenance of a cluster. If the cluster is not short-lived, but users wish to keep on sending queries, it seems more efficient to maintain the group already formed than to create a new one for each set of queries. Further, in light of Section 4, it is clear that user preferences concerning privacy and latency should be appropriately managed by the protocol, particularly when it comes to setting the size parameters m and n.

In general, we wish to implement an efficient protocol for the creation and maintenance of a collaborative structure, in such a way that the privacy protocol is enforced, and in particular, guaranteeing a certain level of robustness against denial-of-service attacks but also accidental network or device failure. A number of techniques in the literature may be exploited to this end.

As an illustrative example, if the privacy protocol is to be used in MANETs, then the creation and maintenance of the trellis structure may reuse existing techniques for cluster management. A survey on clustering in MANETs can be found in [47]. In these clustering protocols, there is no central entity that manages the clusters of the network. Instead, users interact directly in a peer-to-peer fashion. Occasionally, users will act as *cluster heads*, leading the organization of clusters among peers [48, 49]. Since our privacy protocol is not based on any TTP, it seems equally convenient for the ad hoc clustering to be self-organized as well.

5.2. *Untrustworthy Users*

We elaborate on the hypotheses in Section 3.1 regarding selfish, malicious and generally untrustworthy users, and focus our discussion on one of our main scenarios of interest, namely MANETs. Untrustworthy users in MANETs may be simply selfish or intentionally malicious, but may also fail to comply with a particular protocol due to network or device failure. Specifically, selfish users may refrain from carrying out parts of a particular protocol in order to save power, according to the dire energy constraints mobile devices are commonly subject to. Fortunately, reputation systems can frustrate the intentions of selfish users, acting against observable misbehavior to enforce cooperation. In this way, if a node does not behave cooperatively, the affected nodes may decide to deny reciprocal cooperation.

As for dishonest, malicious users, they attempt to compromise the quality of the LBS, perhaps severely. Hence, it is only natural to implement measures to detect and expel them from the collaborative structure [50]. It is clearly in the interest of the service provider and of the rest of users to cooperate to this end. For instance, if a benign user observes inappropriate behavior or fails to receive appropriate responses from another user, it can notify the LBS provider, who may in turn decide to expel the malicious user. Messages passed between parties may be signed to secure the implementation of these countermeasures. One way for the provider to effectively expel users from the service is based on the use of authorization credentials. Precisely, the LBS provider may issue short-lived authorization credentials, such as attribute certificates, to users allowed to use the service. If a user is detected as dishonest or malicious, no further authorization credentials are issued to that user.

A number of reputation systems dealing with user misbehavior have been proposed in the literature. Examples include CORE [51], CONFIDANT [52], SAFE [53] and OCEAN [54]. Further information on trust and reputation systems over ad hoc networks can be found in [55]. Clearly, such systems must be able to make decisions taking into account the realistic possibility of unintended network or device failure, detectable or not.

5.3. *Sybil Attacks*

We now turn to one the hypotheses in Section 3.1, namely the assumption that user IDs may be neither exchanged nor shared. Sybil attacks [56] are those in which an attacker forges an identity, and according to [57], "The Sybil attack can occur in a distributed system that operates without a central authority to verify the identities of each communicating entity." In our setting, a successful Sybil attack means that a single user may act as different users in the cluster, possibly one of them being the cluster head. In that case, the privacy of honest users could be in jeopardy since a single user can collect queries from different users and exert managing responsibilities on the collaborative structure.

As we already mentioned in Section 3.1, our hypotheses essentially preclude Sybil attacks. In practice, this can be accomplished, for example, by enabling parties to authenticate to each other through PKI certificates. In any case, a number of techniques exist in order to counter this type of attacks [57], but a detailed discussion is out of the scope of this work.

5.4. *Confidentiality Attacks and Traffic Analysis*

Finally, we would like to elaborate on the hypothesis in Section 3.1 regarding confidentiality of the communication between users, not only to the LBS provider, and regarding measures to prevent traffic analysis, as defined in Section 2.2. This hypothesis is particularly critical in wireless environments, physically more sensitive to eavesdropping. However, it is not clear that the volatile nature of ad hoc networks could not also play partly in favor of privacy, say by adding uncertainty to an attacker's observations or by making it more difficult to gain control over or information from a number of independent users.

Suppose that malicious users gain access to the content, or even just the length, of the queries received and sent by a target user in the trellis structure of Section 3.3.3. Then these malicious users may be able to ascertain which query was inserted by the target user, and if they collude with the LBS provider, may be able to decipher the contents of the query, or in many cases, at least make certain inferences that may compromise its privacy. Obviously, similar considerations apply to the replies of the LBS.

We sketch a simple cryptographic procedure that helps prevent confidentiality attacks and traffic analysis. The notation $\mathrm{E}_K(M)$ denotes the encryption of a message M with a key K, and (M_1, M_2) denotes the concatenation of messages M_1 and M_2. Suppose that user A in the trellis wishes to send a query Q to user B, through a confidential channel protected with the session key K_{AB}. The sender A also knows the public key P_{LBS} of the LBS provider, and generates a key $K_{\mathrm{A\,LBS}}$ that will be sent to the provider, in order for the provider to encrypt the reply to the query Q. The sender's query is encapsulated in the message

$$\mathrm{A} \rightarrow \mathrm{B}: \ \mathrm{E}_{K_{\mathrm{AB}}}[\mathrm{E}_{P_{\mathrm{LBS}}}(Q, K_{\mathrm{A\,LBS}})].$$

To hinder traffic analysis, assume that all pairs of queries and reply keys $(Q, K_{\mathrm{A\,LBS}})$ fit in a block of common size, for example an RSA cryptogram, commonly after padding. Naturally, a user waits until all queries are received before composing and sending the permuted list containing them, just as mixes do, so that message timing cannot be easily exploited to unveil input-output correspondences. Q may of course be the entire query or actually denote the portion of the query we wish to keep private, such as the user coordinates. Provided that the pairs $(Q, K_{\mathrm{A\,LBS}})$ are sufficiently small, conceivably the entire list of queries sent to the same receiver could be included in a single, ciphered message.

Despite the differences highlighted in Section 3.4, just as with mix networks, our protocol may be vulnerable to, but also benefit from the solutions proposed against, many of the attacks studied reviewed in Section 2.3. An extensive treatment could very well be the objective of future research. Nevertheless, further to the above ideas, we would like to continue our preliminary discussion by pointing out the differences and similarities between attacks against mix networks and against our protocol.

Differently from mix networks, we introduce forged queries that in principle may be constructed to disguise the probabilistic peculiarities of a user's preferences with respect to the overall population, both in terms of locations and questions asked. We saw in Section 4.1 that these forged queries enhance the overall privacy of all the users in the trellis they propagate through. However, similarly to mix networks, there is an inherent, complex trade-off between the degree of anonymity attained, and parameters such as delay, population volume and activity, and traffic overhead. We shall analyze this trade-off from the perspective of quantitative criteria in Section 4. Bear in mind that the determination of some of the limitations of mix networks in the literature may be regarded also as a countermeasure against an attack, in the sense that it provides a design guideline or boundary. For example, [40] analyzes attacks whose success depends on the volume of the population of senders and receivers, the number of receivers a sender communicates with, and the number of communications observed by an attacker. In Section 4.1, we define a probability of privacy breach that depends on the number of simultaneously malicious users. Clearly, sophisticated statistical attacks against mix networks such as [42] are feasible against our protocol as well, a fact that seems to support the intuition that in any system, observability reduces anonymity, to a degree that depends on the complexity or cost of the attack and the anonymity defenses.

6. Conclusion

LBSs are undoubtedly essential representatives of the ICTs. Due to their inherent capability to infer private information from LBSs users, techniques to protect the user privacy are of paramount importance. In this work, we have proposed

a collaborative privacy protocol for LBSs that despite not requiring TTPs, is highly scalable in terms of privacy risk.

A main motivation for our protocol is to minimize infrastructure requirements by excluding the involvement of TTPs such as anonymizers or mix networks, aside from a PKI. In our work, users are *not* assumed to act as mixes, in the sense that while they collect, rearrange and forward messages, they are not assumed to be trustworthy. That is in fact one of the reasons for requiring query forgery. On the other hand, it is only fair to acknowledge that threat models for mix networks do consider the possibility of attackers who can compromise a fraction of the mixes, and the principle of query forgery for query protection is by no means new. In any case, the goal of our protocol is to anonymize a location-based query completely against the recipient itself, namely the LBS provider, whereas mix networks strive to prevent traffic analysis aimed at tracing the connection between senders and receivers. In other words, we design a structure for private retrieval of LBS information, composed only by collaborating albeit untrusted users, endowed with various convenient degrees of freedom regarding the trade-off between privacy, traffic overhead and latency.

Precisely, another of the main strengths of the protocol is that it benefits from an exponentially decreasing probability of guaranteed privacy breach, at the expense of only linearly increasing relative communication costs, with respect to the size parameters of the trellis.

More specifically, users group themselves into a trellis of m rows and n columns, where queries are exchanged and permuted in such a way that privacy is preserved throughout to a scalable degree. In fact, complete privacy breach is only guaranteed under the collusion of $2m$ users together with the LBS provider, increasingly unlikely with large m . Assuming that users act maliciously with probability $1 - t$ independently from each other, conditioned on the event that the provider is also malicious, the probability of guaranteed privacy breach is given by the exponential expression $(1 - t)^{2m}$.

Regarding network costs, the number of connections and query messages required by our protocol in the m-by-n trellis is $O(m^2\, n)$, and the total number of queries transmitted through the trellis is $O(m\, n^2)$. Relative to the minimum of $m\, n$ attained in an ideal scenario with nonmalicious participants, the corresponding overhead is linear, precisely, $O(m)$ and $O(n)$ respectively.

Provided that users in the first column generate a fixed number of queries, the total number of queries processed by the LBS is $O(m\, n)$, and the relative overhead is asymptotically 1. We describe simple variations of the protocol to remove the need for forged query processing altogether. In addition, the privacy protocol is completely transparent to the implementation of the query-response function in the LBS. This is an advantage with respect to cryptographic PIR mechanisms operating on the assumption of a reduced response space, for instance, a lookup table implementation of the query-response function.

There exists a trade-off between privacy and latency, due to the fact that users must wait for others to cooperate before sending their queries, and that a latency constraint in turn imposes un upper bound on the average number of participants in the trellis.

Creating and maintaining the ad hoc network structure needed for our protocol has been shown to be feasible with a small number of users. This may be sufficient in practical applications because our proposal does not need a large number of participants, due to the exponentially low likelihood of privacy breach. However, an interesting challenge arises from the fact that the protocol may be improved by devising a completely secure and more efficient mechanism to create and maintain collaborative structures, and to enforce the privacy protocol presented, in particular against denial of service attacks, and for large-scale structures.

Another future research direction arises naturally from the acknowledgment that sophisticated statistical attacks against mix networks are feasible against our protocol as well, a fact that seems to support the intuition that in any system, observability reduces anonymity, to a degree that depends on the complexity or cost of the attack and of the anonymity defenses.

Even though the main motivation of this work is LBSs, our protocol is, in principle, applicable beyond LBSs to any sort of PIR, in the sense that the entire query content is anonymized, not only the location information. An example of application, alternative to LBSs, is private Internet search. In other words, no requirements are made regarding the properties, mathematical or otherwise, of the query information protected, nor is this information perturbed in any way. This is an advantage with respect to other protocols for privacy in LBSs that assume, for example, that a set of locations can be added, and consequently exploit cryptographic homomorphisms, or send perturbed or even ambiguous location information, such as a region rather than a point.

Acknowledgment and Disclaimer

We gratefully acknowledge the valuable comments of Claudia Díaz from the Katholieke Universiteit Leuven. We would also like to thank the anonymous reviewers for thorough, insightful comments that greatly improved the manuscript. This work was partly supported by the Spanish Government through projects CONSOLIDER INGENIO 2010 CSD2007-00004 “ARES”, TSI2007-65393-C02-02 “ITACA” and TSI2007-65406-C03-01 “E-AEGIS”, and by the Government of Catalonia under grants 2009 SGR 1135 and 2009 SGR 1362.

The material in this paper was presented in part at the IADIS Conference on the e-Society, Barcelona, Spain, Feb. 2009 [58].

Appendix A. Probability of Coincidental Complete Privacy Breach

We prove the mathematical assertions on the probability of coincidental complete privacy breach in the paper, specifically, (1) and (2) in Section 4.1.2.

By symmetry, the probability of CCPB for user (i,j) satisfies,

$$\begin{aligned}\mathrm{P}\{\mathrm{CCPB}\} &= \sum_{i'=1}^{m} \mathrm{P}\{\mathrm{CCPB}|\text{query sent to } (i',j+1)\} \\ &\quad \times \mathrm{P}\{\text{query sent to } (i',j+1)\} \\ &= m\,\mathrm{P}\{\mathrm{CCPB}|\text{query sent to } (1,j+1)\}\frac{1}{m} \\ &= \mathrm{P}\{\mathrm{CCPB}|\text{query sent to } (1,j+1)\}, \quad \text{(A.1)}\end{aligned}$$

where the event "query sent to $(i',j+1)$" indicates that the user of interest sent their own query to $(i',j+1)$. Thus without loss of generality we may assume that (i,j) sent their query to user $(1,j+1)$. Note that if $(1,j+1)$ were not malicious, their query and (i,j)'s would be undistinguishable by a privacy attacker.

Let $j>1$. Denote by the r.v.'s $B_1,\dots B_{j-1}$ the number of benign users in the respective columns, taking values between 0 and m. Consider the event that $B_1 = b_1,\dots,B_{j-1}=b_{j-1}$, which occurs with probability

$$\left(\prod_{k=1}^{j-1}\binom{m}{b_k}\right) t^{\sum_{k=1}^{j-1} b_k}\,(1-t)^{m\,(j-1)-\sum_{k=1}^{j-1} b_k}.$$

We now compute the probability, conditioned on this event, that no benign user in column $k=1,\dots,j-1$ sends their query to (i,j), such that it remains unseen by malicious users. The assumption of random query forwarding yields the probability $(1-\frac{1}{m})^{b_{j-1}}$ in the special case when $k=j-1$. For $k=j-2$, this probability becomes $\left(1-\frac{b_j-1}{m}\frac{1}{m}\right)^{b_{j-2}}$. More generally, it is $\left(1-\frac{\prod_{l=k+1}^{j-1} b_l}{m^{j-k}}\right)^{b_k}$ for $k>1$, and $\left(1-\frac{\prod_{k=2}^{j-1} b_k}{m^{j-1}}\right)^{b_1\,(1+f)}$ for $k=1$. The probability of CCPB given the number of benign users in each column is the probability that user $(1,j+1)$ is malicious, and no benign user in columns 1 through $j-1$ sends their query to (i,j), such that it is unobserved by any malicious users. Precisely, it is

$$(1-t)\left(1-\frac{\prod_{k=2}^{j-1} b_k}{m^{j-1}}\right)^{b_1\,(1+f)} \times \prod_{k=2}^{j-1}\left(1-\frac{\prod_{l=k+1}^{j-1} b_l}{m^{j-k}}\right)^{b_k}.$$

Application of the previous results to

$$p_{\mathrm{CCPB}} = \sum_{b_1=0}^{m}\cdots\sum_{b_{j-1}=0}^{m} \mathrm{P}\{\mathrm{CCPB}|b_1,\dots,b_{j-1}\} \times \mathrm{P}\{b_1,\dots,b_{j-1}\}$$

immediately gives (1).

We now prove the bounds (2), which, interestingly, reflect the behavior of $(1,j+1)$ and users in column $j-1$ only. The lower bounds follows immediately from (A.1) and the fact that CCPB is achieved in particular when $(1,j+1)$ and all users in column $j-1$ are malicious. The probability that a user in column $j-1$ is benign and sends their query to (i,j) is t/m by symmetry. Hence, $(1-t/m)^m$ is the probability that all users in column $j-1$ are either malicious or do not send their query to (i,j). CCPB requires this outcome, and that user $(1,j+1)$ be malicious. This establishes the upper bound.

Finally, to compute the first-order Taylor approximation (3), we may neglect all but the term of (1) corresponding to $b_1=\cdots=b_{j-1}=m$, because they contain higher powers of $1-t$. But the first-order approximation to a function of the form $(1-t)f(t)$ at 1 (for any differentiable f) is simply $(1-t)f(1)$. (In fact, regardless of whether random query forwarding is enforced, $p_{\mathrm{CCPB}} = (1-t)\,\mathrm{P}\{\mathrm{CCPB}|m,\dots,m\}$.)

References

[1] K. Allison, R. Minto, 3G iPhone sales hit 1m during opening weekend, Financial Times.
URL http://www.ft.com/cms/s/0/9d4ea864-51cc-11dd-a97c-000077b07658.html

[2] S. Benford, C. Magerkurth, P. Ljungstrand, Bridging the physical and digital in pervasive gaming, Commun. ACM 48 (3) (2005) 54–57.

[3] M. M. Tsang, S. C. Ho, T. P. Liang, Consumer attitudes toward mobile advertising: An empirical study, Int. J. Electron. Commer. 8 (3) (2004) 65–78.

[4] L. Bauer, L. F. Cranor, R. W. Reeder, M. K. Reiter, K. Vaniea, A user study of policy creation in a flexible access-control system, in: Procc SIGCHI Conf. Hum. Fact. Comput. Syst., ACM, Florence, Italy, 2008, pp. 543–552.

[5] W3C, Platform for privacy preferences (P3P) project (Oct. 2007).
URL http://www.w3.org/P3P

[6] H. Schulzrinne, H. Tschofenig, J. Morris, J. Cuellar, J. Polk, Geolocation policy, Tech. rep., Internet Eng. Task Force (Jun. 2008).
URL http://www.ietf.org/internet-drafts/draft-ietf-geopriv-policy-17.txt

[7] D. Chaum, Security without identification: Transaction systems to make big brother obsolete, Commun. ACM 28 (10) (1985) 1030–1044.

[8] V. Benjumea, J. López, J. M. T. Linero, Specification of a framework for the anonymous use of privileges, Telemat., Informat. 23 (3) (2006) 179–195.

[9] G. Bianchi, M. Bonola, V. Falletta, F. S. Proto, S. Teofili, The SPARTA pseudonym and authorization system, Sci. Comput. Program. 74 (1–2) (2008) 23–33.

[10] M. Gruteser, D. Grunwald, Anonymous usage of location-based services through spatial and temporal cloaking, in: Proc. ACM Int. Conf. Mob. Syst., Appl., Serv. (MobiSys), ACM, San Francisco, CA, 2003, pp. 31–42.

[11] M. Duckham, K. Mason, J. Stell, M. Worboys, A formal approach to imperfection in geographic information, Comput., Environ., Urban Syst. 25 (1) (2001) 89–103.

[12] M. Duckham, L. Kulit, A formal model of obfuscation and negotiation for location privacy, in: Proc. Int. Conf. Pervas.

Comput., Vol. 3468 of Lecture Notes Comput. Sci. (LNCS), Springer-Verlag, Munich, Germany, 2005, pp. 152–170.

[13] C. A. Ardagna, M. Cremonini, E. Damiani, S. De Capitani di Vimercati, P. Samarati, Location privacy protection through obfuscation-based techniques, in: Proc. Annual IFIP Working Conf. Data Appl. Security, Vol. 4602 of Lecture Notes Comput. Sci. (LNCS), Springer-Verlag, Redondo Beach, CA, 2007, pp. 47–60.

[14] M. L. Yiu, C. S. Jensen, X. Huang, H. Lu, SpaceTwist: Managing the trade-offs among location privacy, query performance, and query accuracy in mobile services, in: Proc. IEEE Int. Conf. Data Eng. (ICDE), Cancun, Mexico, 2008, pp. 366–375.

[15] C. Chow, M. F. Mokbel, X. Liu, A peer-to-peer spatial cloaking algorithm for anonymous location-based services, in: Proc. ACM Int. Symp. Adv. Geogr. Inform. Syst. (GIS), Arlington, VA, 2006, pp. 171–178.

[16] P. Samarati, L. Sweeney, Protecting privacy when disclosing information: k-Anonymity and its enforcement through generalization and suppression, Tech. rep., SRI Int. (1998).

[17] P. Samarati, Protecting respondents' identities in microdata release, IEEE Trans. Knowl. Data Eng. 13 (6) (2001) 1010–1027.

[18] J. Domingo-Ferrer, Microaggregation for database and location privacy, in: Proc. Int. Workshop Next-Gen. Inform. Technol., Syst. (NGITS), Vol. 4032 of Lecture Notes Comput. Sci. (LNCS), Springer-Verlag, Kibbutz Shefayim, Israel, 2006, pp. 106–116.

[19] A. Solanas, A. Martínez-Ballesté, A TTP-free protocol for location privacy in location-based services, Comput. Commun. 31 (6) (2008) 1181–1191.

[20] G. Ghinita, P. Kalnis, A. Khoshgozaran, C. Shahabi, K.-L. Tan, Private queries in location based services: Anonymizers are not necessary, in: Proc. ACM SIGMOD Int. Conf. Manage. Data, Vancouver, Canada, 2008, pp. 121–132.

[21] R. Ostrovsky, W. E. Skeith III, A survey of single-database PIR: Techniques and applications, in: Proc. Int. Conf. Practice, Theory Public-Key Cryptogr. (PKC), Vol. 4450 of Lecture Notes Comput. Sci. (LNCS), Springer-Verlag, Beijing, China, 2007, pp. 393–411.

[22] M. F. Mokbel, Towards privacy-aware location-based database servers, in: Proc. IEEE Int. Conf. Data Eng. Workshops (PDM), Atlanta, GA, 2006, p. 93.

[23] B. Gedik, L. Liu, A customizable k-anonymity model for protecting location privacy, in: Proc. IEEE Int. Conf. Distrib. Comput. Syst. (ICDS), Columbus, OH, 2005, pp. 620–629.

[24] R. Cheng, Y. Zhang, E. Bertino, S. Prabhakar, Preserving user location privacy in mobile data management infrastructures, in: Proc. Workshop Privacy Enhanc. Technol. (PET), Vol. 4258 of Lecture Notes Comput. Sci. (LNCS), Springer-Verlag, Cambridge, United Kingdom, 2006, pp. 393–412.

[25] B. Gedik, L. Liu, Protecting location privacy with personalized k-anonymity: Architecture and algorithms, IEEE Trans. Mob. Comput. 7 (1) (2008) 1–18.

[26] B. Bamba, L. Liu, P. Pesti, T. Wang, Supporting anonymous location queries in mobile environments with PrivacyGrid, in: Proc. Int. World Wide Web Conf. (WWW), Beijing, China, 2008, pp. 237–246.

[27] B. Hoh, M. Gruteser, H. Xiong, A. Alrabady, Enhancing security and privacy in traffc-monitoring systems, IEEE Pervas. Comput. Mag., Special Issue Intell. Transport. Syst. 5 (4).

[28] G. Zhong, U. Hengartner, A distributed k-anonymity protocol for location privacy, in: Proc. IEEE Int. Conf. Pervas. Comput., Commun. (PerCom), Galveston, TX, 2009, pp. 253–262.

[29] C. Gülcü, G. Tsudik, Mixing email with Babel, in: Proc. IEEE Symp. Netw. Distrib. Syst. Security (SNDSS), Washington, DC, 1996, pp. 2–16.

[30] G. Danezis, R. Dingledine, N. Mathewson, Mixminion: Design of a type III anonymous remailer protocol, in: Proc. IEEE Symp. Security, Privacy (SP), Berkeley, CA, 2003, pp. 2–15.

[31] G. Danezis, Statistical disclosure attacks: Traffic confirmation in open environments, in: Proc. Security, Privacy, Age Uncertainty, (SEC), Athens, Greece, 2003, pp. 421–426.

[32] M. J. Freedman, R. Morris, Tarzan: A peer-to-peer anonymizing network layer, in: Proc. ACM Conf. Comput., Commun. Security (CCS), Washington, DC, 2002.

[33] M. J. Freedman, E. Sit, J. Cates, R. Morris, Introducing Tarzan, a peer-to-peer anonymizing network layer, in: Proc. ACM Conf. Comput., Commun. Security (CCS), Washingtion, DC, 2003, pp. 193–206.

[34] M. G. Reed, P. F. Syverson, D. M. Goldschlag, Proxies for anonymous routing, in: Proc. Comput. Security Appl. Conf. (CSAC), San Diego, CA, 1996, pp. 9–13.

[35] D. Goldschlag, M. Reed, P. Syverson, Onion routing, Commun. ACM 42 (2).

[36] R. Dingledine, N. Mathewson, P. Syverson, TOR: The second-generation onion router, in: Proc. Conf. USENIX Security Symp., Berkeley, CA, 2004.

[37] O. Berthold, A. Pfitzmann, R. Standtke, The disadvantages of free MIX routes and how to overcome them, in: Proc. Design. Privacy Enhanc. Technol.: Workshop Design Issues Anon., Unobser., Lecture Notes Comput. Sci. (LNCS), Springer-Verlag, Berkeley, CA, 2000, pp. 30–45.

[38] G. Danezis, C. Díaz, A survey of anonymous communication channels, Tech. Rep. MSR-TR-2008-35, Microsoft Res. (Jan. 2008).

[39] D. Chaum, Untraceable electronic mail, return addresses, and digital pseudonyms, Commun. ACM 24 (2) (1981) 84–88.

[40] D. Kesdogan, D. Agrawal, S. Penz, Limits of anonymity in open environments, in: Proc. Inform. Hiding Workshop (IH), Lecture Notes Comput. Sci. (LNCS), Springer-Verlag, Noordwijkerhout, The Netherlands, 2002.

[41] C. Troncoso, B. Gierlichs, B. Preneel, I. Verbauwhede, Perfect matching disclosure attacks, in: Proc. Workshop Privacy Enhanc. Technol. (PET), Vol. 5134 of Lecture Notes Comput. Sci. (LNCS), Springer-Verlag, Leuven, Belgium, 2008, pp. 2–23.

[42] G. Danezis, C. Troncoso, Vida: How to use bayesian inference to de-anonymize persistent communications, in: Proc. Workshop Privacy Enhanc. Technol. (PET), Seattle, WA, 2009.

[43] J. Forné, J. L. Muñoz, X. Hinarejos, O. Esparza, Certificate status validation in mobile ad hoc networks, IEEE Wirel. Commun. Mag. 16 (1) (2009) 55–62.

[44] N. Li, T. Li, S. Venkatasubramanian, t-Closeness: Privacy beyond k-anonymity and l-diversity, in: Proc. IEEE Int. Conf. Data Eng. (ICDE), Istanbul, Turkey, 2007, pp. 106–115.

[45] D. Rebollo-Monedero, J. Forné, J. Domingo-Ferrer, From t-closeness to PRAM and noise addition via information theory, in: Privacy Stat. Databases (PSD), Lecture Notes Comput. Sci. (LNCS), Springer-Verlag, Istambul, Turkey, 2008, pp. 100–112.

[46] C. Díaz, Anonymity and privacy in electronic services, Ph.D. thesis, Katholieke Univ. Leuven (Dec. 2005).

[47] J. Yu, P. H. J. Chong, A survey of clustering schemes for mobile and ad hoc networks, IEEE Commun. Surv., Tutor. 7 (1) (1999) 32–48.

[48] C. Cramer, O. Stanze, K. Weniger, M. Zitterbart, Demand-driven clustering in MANETs, in: Proc. Int. Conf. Wirel. Netw. (ICWN), Vol. 1, Las Vegas, NV, 2004, pp. 81–87.

[49] N. Chatterjee, A. Potluri, A. Negi, A scalable and adaptive clustering scheme for MANETs, in: Proc. Int. Conf. Distrib. Comput., Internet Technol. (ICDCIT), Vol. 4882 of Lecture Notes Comput. Sci. (LNCS), Springer-Verlag, Bangalore, India, 2007, pp. 73–78.

[50] A. Srinivasan, J. Teitelbaumy, H. Liangz, J. Wuyand, M. Cardei, Algorithms and Protocols for Wireless Ad Hoc and Sensor Networks, John Wiley & Sons, NJ, 2007, Ch. Reputation and Trust-Based Systems for Ad Hoc and Sensor Networks.

[51] P. Michiardi, R. Molva, CORE: A COllaborative REputation mechanism to enforce node cooperation in mobile ad hoc

networks, in: Proc. IFIP Commun., Multimedia Security Conf., Portoroz, Slovenia, 2002, pp. 107–121.

[52] S. Buchegger, J. Y. Le Boudec, Performance analysis of the CONFIDANT protocol, in: Proc. ACM Int. Symp. Mob. Ad Hoc Netw., Comput. (MobiHoc), Lausanne, Switzerland, 2002, pp. 226–236.

[53] Y. Rebahi, V. Mujica, C. Simons, D. Sisalem, SAFE: Securing pAcket Forwarding in ad hoc nEtworks, in: Proc. Workshop Appl., Serv. Wirel. Netw., Paris, France, 2005.

[54] S. Bansal, M. Baker, Observation based cooperation enforcement in ad hoc networks, Tech. rep., Stanford Univ. (2003).

[55] M. Mejía, N. Peña, J. L. Muñoz, O. Esparza, A review for trust modelling in ad hoc networks, Internet Research 19 (1) (2009) 88–104.

[56] J. Newsome, E. Shi, D. Song, A. Perrig, The Sybil attack in sensor networks: Analysis and defenses, in: Proc. Int. Symp. Inform. Processing Sensor Netw. (IPSN), Berkeley, CA, 2004, pp. 259–268.

[57] C. Piro, C. Shields, B. N. Levine, Detecting the Sybil attack in mobile ad hoc networks, in: Proc. Int. Conf. Security, Privacy Commun. Netw. (SecureComm), Baltimore, MD, 2006, pp. 81–87.

[58] D. Rebollo-Monedero, J. Forné, L. Subirats, A. Solanas, A. Martínez-Ballesté, A collaborative protocol for private retrieval of location-based information, in: Proc. IADIS Int. Conf. e-Society, Barcelona, Spain, 2009.