Claudio Bettini
Sushil Jajodia
Pierangela Samarati
X. Sean Wang

PiLBA’08

Privacy in Location-Based Applications

Workshop co-located with ESORICS 2008

Malaga, Spain, October 9, 2008
Proceedings

Editors' addresses:

Claudio Bettini
D.I.Co. , Universitá di Milano - Via Comelico 39, I-20135 Milano, Italy
bettini@dico.unimi.it

Sushil Jajodia
C.S.I.S. , George Mason University - 4400 University Drive, Fairfax, VA 22030, USA
jajodia@gmu.edu

Pierangela Samarati
D.T.I. , Universitá di Milano - Via Bramante 65 , 26013 Crema , Italy
samarati@dti.unimi.it

X. Sean Wang
Dep. of C.S. , University of Vermont - 33 Colchester Avenue, Burlington, VT 05405, USA
xywang@cs.uvm.edu

Preface

Location based applications in travel, logistics, health care, social networks and other industries already exist and are poised to proliferate. One of the critical issues for a widespread deployment of these applications is how to conciliate their effectiveness and quality with privacy concerns. The PiLBA '08 workshop was intended to bring together scientists from security and data management to discuss the most recent advances in the field. These proceedings include the eight contributions on this topic selected from the submissions by the PC chairs with the help of the members of the program committee, and presented at the workshop. They include an extended abstract of the invited talk, two survey papers, and five research papers covering several complementary aspects of privacy in location based applications and services.

The Organizing Committee would like to thank all the people that supported and helped in the organization of this event. A particular acknowledgment to the PC members, to Javier Lopez for his suggestions and support with local organization and to Linda Pareschi for her help with the web site and with the preparation of these proceedings.

Claudio Bettini
Sushil Jajodia
Pierangela Samarati
X. Sean Wang

October 2008

Organizing Committee

Claudio Bettini - Universitá di Milano, Italy
Sushil Jajodia - George Mason University, USA
Pierangela Samarati - Universitá di Milano, Italy
Sean Wang - University of Vermont, USA

Publicity Chair

Linda Pareschi - Universitá di Milano, Italy

Programme Committee

Claudio Agostino Ardagna - Universitá di Milano, Italy
Vijay Atluri - Universitá di Milano, Italy
Ying Cai - Iowa State University, USA
Reynold Cheng - University of Hong Kong, China
Josep Domingo-Ferrer - Rovira i Virgili University, Spain
Marco Gruteser - Rutgers University, USA
Urs Hengartner - University of Waterloo, Canada
Panos Kalnis - National University of Singapore, Singapore
George Kollios - Boston University, USA
Ling Liu - Georgia Institute of Technology, USA
Sergio Mascetti - Universitá di Milano, Italy
Dino Pedreschi - Universitá di Pisa, Italy
Daniele Riboni - Universitá di Milano, Italy
Cyrus Shahabi - University of Southern California, USA
Heng Xu - Penn State University, USA
Man Lung Yiu - Aalborg University, Denmark

External Reviewers

Baik Hoh
Ali Khoshgozaran
Houtan Shirani-Mehr
Xike Xie
Yan Zhang

Contents

Safety and Privacy in Vehicular Communications

Josep Domingo-Ferrer and Qianhong Wu

Universitat Rovira i Virgili, UNESCO Chair in Data Privacy, Dept. of Computer
Engineering and Mathematics, Av. Països Catalans 26, E-43007 Tarragona, Catalonia
e-mail {josep.domingo,qianhong.wu}@urv.cat

Abstract. Vehicular *ad hoc* networks (VANETs) allow vehicles to disseminate messages about road conditions to other vehicles. As long as these messages are trustworthy, they can greatly increase traffic safety and efficiency. Hence, care must be exerted to ensure that vehicle-generated messages do not convey inaccurate or false content. A natural way to proceed is to request endorsement by nearby vehicles on the content of a message originated by a certain vehicle. However, such a message generation and peer-to-peer endorsement should not result in any privacy loss on the part of vehicles co-operating in it. We survey the available solutions to this security-privacy tension and discuss their limitations. We sketch a new privacy-preserving system which guarantees message authentication through both *a priori* and *a posteriori* countermeasures.

1 Introduction

VANETs allow vehicles to broadcast messages to other vehicles in the vicinity. It is suggested that each vehicle periodically send messages over a single hop every 300ms within a distance of 10s travel time (which is a distance range between 10 m and 300 m)[RH05]. This mechanism can be used to improve safety and optimize traffic. However, malicious vehicles can also make use of this mechanism by sending fraudulent messages for their own profit or just to jeopardize the traffic system. Hence, the system must be designed to ensure that the transmission comes from a trusted source and has not been tampered with since transmission.

Another critical concern in VANETs is driving privacy or vehicle anonymity. As noted in [Dot06], a lot can be inferred on the driver's privacy if the whereabouts and the driving pattern of a car can be tracked. However, it is possible for attackers to trace vehicles by using cameras or physical tracking. But such physical attacks can only trace specific targets and are much more expensive than monitoring the communication in VANETs. This paper addresses the latter attacks.

2 Countermeasures for securing VANETs

VANETs function to improve safety only if the messages sent by vehicles are trustworthy. Dealing with fraudulent messages is a thorny issue for safety engineers due to the self-organized property of VANETs. The situation is deteriorated by the privacy requirements of vehicles since, in a privacy-preserving setting, the message generators, *i.e.* the vehicles, are anonymous in the sense that their identities are unknown. A number of schemes have been proposed to reduce fraudulent messages; such proposals fall into two classes, namely *a posteriori* and *a priori*.

2.1 A posteriori countermeasures

A posteriori countermeasures consist of taking punitive action against vehicles who have been proven to have originated fraudulent messages. To be compatible with privacy preservation, these countermeasures require the presence of a trusted third party able to open the identities of dishonest vehicles. Then the revoked vehicles can be expelled from the system. Cryptographic authentication technologies have been extensively exploited to offer *a posteriori* countermeasures. Some proposals use regular digital signatures [RPH06,RH07,RPAJ07,AFWZ07]. In these proposals, vehicle privacy is provided by a pseudonym mechanism, in which certificate authorities (CAs) produce many pseudonyms for each vehicle so that attackers cannot trace the vehicles producing signatures in different periods with different pseudonyms, except if the CAs open the identities of the vehicles. The pseudonym mechanism is not that efficient due to the heavy overhead of pseudonym generation and storage. Other schemes use sophisticated cryptographic technologies such as group signatures [GBW07] or ring signatures [LSHS07,GGT06]. The latter methods are more efficient, but those using ring signatures cannot trace malicious vehicles due to the unconditional anonymity of ring signatures. Along this research line, the scheme in [GBW07] seems the most efficient one that can provide revokable anonymity.

2.2 A priori countermeasures

A priori countermeasures attempt to prevent the generation of fraudulent messages. This approach is based on the assumption that most users are honest and will not endorse any message containing false data. Another implicit assumption is the usual common sense that, the more people endorse a message, the more trustworthy it is. Along this research line, the schemes in [GGS04,ODS07,PP05,RAH06,DDSV08] exploit the assumption that there is a majority of honest vehicles in VANETs. Hence, these schemes introduce some form of threshold mechanism: a message is trusted if it has been verifiably endorsed by a number of vehicles above a certain threshold. Among these schemes, the proposals in [DDSV08] may be the most efficient while enabling anonymity of message originators. But their scheme does not provide anonymity revocability, which may not suit some applications in which anonymity must be revoked "for the prevention, investigation, detection and prosecution of serious criminal offences"[EP05].

3 Discussion

Unfortunately, neither *a posteriori* nor *a priori* countermeasures are solely sufficient to secure VANETs. By taking strict punitive action, *a posteriori* countermeasures can exclude some rational attackers producing bogus messages to obtain benefits or pranks. However, they are ineffective against irrational attackers such as terrorists. Even for rational attackers, damage has already occurred when punitive action is taken. It seems that *a priori* countermeasures function better in this case because they prevent damage beforehand by letting the vehicles trust only messages endorsed by a number of vehicles. Although the underlying assumption that there is a majority of honest vehicles in VANETs generally holds, it cannot be excluded that a number of malicious vehicles greater than or equal to the threshold are present in specific locations, for instance. For example, this is very plausible if some criminal organization undertakes to divert traffic from a certain area by broadcasting messages informing that a road is barred. Furthermore, for convenience in implementation, existing schemes use an even stronger assumption that the number of honest vehicles in all cases should be at least a preset threshold. But such a universally valid threshold does not exist in practice. Indeed, the threshold should somehow take the traffic density and the message scope into account: a low density of vehicles calls for a lower threshold, whereas a high density and a message relevant to the whole traffic of a city requires a sufficiently high threshold.

The situation is aggravated by the anonymity technologies used some proposals. A system preserves anonymity when it does not require the identity of its users to be disclosed. Without anonymity, attackers can trace all the vehicles by monitoring the communication in VANETs, which in turn can enable the attackers to mount serious attacks against specific targets. Hence, anonymity is a critical concern in VANETs. However, anonymity can also weaken *a posteriori* and *a priori* countermeasures. Indeed, attackers can send fraudulent messages without fear of being caught due to anonymity, and as a result, no punitive action can be taken against them. Furthermore, some proposals provide strong anonymity, *i.e.* unlinkability. Unlinkability implies that a verifier cannot distinguish whether two signatures come from the same vehicle or two vehicles. This feature may enable malicious vehicles to mount the so-called Sybil attack: a vehicle generates a fraudulent message and then endorses the message herself by computing on it as many signatures as required by the threshold in use; since signatures are unlinkable, no one can find out that all of them come from the same vehicle. Hence, elegantly designed protocols are required to secure VANETs when incorporating anonymity.

4 Towards a combination of *a priori* and *a posteriori* countermeasures

Bearing in mind that enhancing safety and traffic efficiency is one of the main thrusts behind VANETs, we propose a new efficient system to balance public safety and vehicle privacy. Both *a priori* and *a posteriori* countermeasures are resorted to in order to thwart attackers. To the best of our knowledge, ours is the first system equipped with both types of countermeasures. We achieve this goal by drawing on the novel technology of message-linkable group signatures (MLGS). In an MLGS scheme, a vehicle stays anonymous if it produces two signatures on two different messages. However, if it produces two signatures on the same message, then it will be identified, which effectively thwarts the Sybil attack in a privacy-preserving system. This novel technology also enables us to realize a threshold-adaptive authentication in which the threshold can adaptively change in light of the context of messages, instead of having to be preset during the system design stage. Furthermore, a fast batch verification method is presented to speed up the validation of authenticated messages. Since vehicles periodically receive a large number of messages to be validated, such a batch verification is critical to make authentication implementable in VANETs. Details on the new scheme will be given in [WD08].

Acknowledgments and disclaimer

This work was partly supported by the Spanish Government through projects CONSOLIDER INGENIO 2010 CSD2007-00004 “ARES” and

TSI2007-65406-C03-01 “E-AEGIS”, and by the Government of Catalonia under grant 2005 SGR 00446. The authors are with the UNESCO Chair in Data Privacy, but their views do not necessarily reflect the position of UNESCO nor commit that organization.

References

[RH05] M. Raya and J.-P. Hubaux. The security of vehicular ad hoc networks. In SASN'05, 2005.

[Dot06] F. Dötzer. Privacy issues in vehicular ad hoc networks. Lecture Notes in Computer Science, vol. 3856, pp. 197-209, 2006.

[RPH06] M. Raya, P. Papadimitratos and J.-P. Hubaux. Securing vehicular communications. IEEE Wireless Communications Magazine, vol. 13, no. 5, pp. 8-15, 2006.

[RH07] M. Raya and J.-P. Hubaux. Securing vehicular ad hoc networks. Journal of Computer Security, Special Issue on Security of Ad Hoc and Sensor Networks, vol. 15, no. 1, pp. 39-68, 2007.

[RPAJ07] M. Raya, P. Papadimitratos, I. Aad, D. Jungels and J.-P. Hubaux. Eviction of misbehaving and faulty nodes in vehicular networks. IEEE Journal on Selected Areas in Communications, vol. 25, no. 8, pp. 1557-1568, 2007.

[AFWZ07] F. Armknecht, A. Festag, D. Westhoff and K. Zeng. Cross-layer privacy enhancement and non-repudiation in vehicular communication. In 4th Workshop on Mobile Ad-Hoc Networks (WMAN), Bern, Switzerland, March 2007.

[GBW07] J. Guo, J.P. Baugh and S. Wang. A group signature based secure and privacy-preserving vehicular communication framework. In Mobile Networking for Vehicular Environments, pp. 103-108, 2007.

[LSHS07] X. Lin, X. Sun, P.-H. Ho and X. Shen. GSIS: A secure and privacy preserving protocol for vehicular communications. IEEE Transactions on Vehicular Technology, vol. 56, no. 6, pp. 3442-3456, 2007.

[GGT06] C. Gamage, B. Gras and A.S. Tanenbaum. An identity-based ring signature scheme with enhanced privacy. In Proceedings of the IEEE SecureComm Conference, pp. 1-5, 2006.

[GGS04] P. Golle, D. Greene and J. Staddon. Detecting and correcting malicious data in VANETs. In Proceedings of the 1st ACM international workshop on Vehicular Ad Hoc Networks, pp. 29-37, 2004.

[PP05] B. Parno and A. Perrig. Challenges in securing vehicular networks. In Proceedings of the ACM Workshop on Hot Topics in Networks, 2005.

[ODS07] B. Ostermaier, F. Dötzer and M. Strassberger. Enhancing the security of local danger warnings in VANETs - A simulative analysis of voting schemes. In Proceedings of the Second International Conference on Availability, Reliability and Security, pp. 422-431, 2007.

[RAH06] M. Raya, A. Aziz and J.-P. Hubaux. Efficient secure aggregation in VANETs. In Proceedings of the 3rd International Workshop on Vehicular Ad hoc Networks - VANET 06, pp. 67-75, 2006.

[DDSV08] V. Daza, J. Domingo-Ferrer, F. Sebe and A. Viejo. Trustworthy privacy-preserving car-generated announcements in vehicular ad hoc networks. IEEE Transactions on Vehicular Technology, Accepted, July 2008.

[EP05] European Parliament. Legislative resolution on the proposal for a directive of the European Parliament and of the Council on the retention of data processed in connection with the provision of public electronic communication services and amending Directive 2002/58/EC (COM(2005)0438 C6-0293/2005 2005/0182(COD)), 2005

[WD08] Q. Wu and J. Domingo-Ferrer. Improved trustworthiness of vehicular communications with a priori and a posteriori countermeasures. Manuscript in preparation, 2008.

Location Privacy in Location-Based Services: Beyond TTP-based Schemes

Agusti Solanas, Josep Domingo-Ferrer, and Antoni Martínez-Ballesté

Rovira i Virgili University
Department of Computer Engineering and Maths
UNESCO Chair in Data Privacy
Av. Països Catalans 26
E-43007 Tarragona, Catalonia, Spain
{agusti.solanas,josep.domingo,antoni.martinez}@urv.cat

Abstract. Location-Based Services (LBS) are gaining importance due to the advances in mobile networks and positioning technologies. Nevertheless, the wide deployment of LBS can jeopardise the privacy of their users, so ensuring user privacy is paramount to the success of those services. This article surveys the most relevant techniques for guaranteeing location privacy to LBS users. The rigid dichotomy between schemes which rely on Trusted Third Parties (TTP-based) and those which do not (TTP-free) is emphasised. Also, the convenience of both approaches is discussed and some ideas on the future of location privacy in these services are sketched.

Keywords: Anonymisation/pseudonymisation in LBS, Trust management in LBS.

1 Introduction

The Information Society rests on the Information and Communications Technologies (ICT). Location-Based Services (LBS) are becoming an important ICT and will be eventually available anywhere anytime. LBS provide users with highly personalised information accessible by means of a variety of mobile devices that are able to locate themselves, e.g. by using a GPS or a fixed network infrastructure with GSM [1]. Mobile devices are ubiquitous and services related to the user's current location proliferate. Examples of LBS are location-based tourist information [2], route guidance [3], emergency assistance [4], location-based advertising [5], etc.

The extensive deployment of ubiquitous technology is not without privacy drawbacks. By sending their locations, LBS users could endanger their security and privacy because, for example, an attacker could determine their location and track them. This tracking capability of attackers opens up many computer-aided crime possibilities (harassment, car theft, kidnapping, etc.). Also, if an attacker impersonates an LBS provider, the traffic patterns of LBS users could be influenced by false information, and the users' location could be compromised [6].

There are also other attacks which aim to identify users by means of the locations contained in their queries. By identifying users, attackers can link queries to real identities. In those ways, attackers can obtain detailed profiles of the users and send them undesired advertisements or even harass them. Some examples of techniques/attacks for identifying users are the restricted space identification (RSI) attack and the observation identification (OI) attack. The RSI attack consists in linking locations to identities by using queries which are submitted from a restricted space (e.g. if a user submits queries from his garage in a suburban house, it is easy to link those queries to his real identity by looking up who lives in that house, for example by means of a phonebook). Similarly the OI attack links queries to identities by observing where users are (i.e. the attacker knows the user's location because she can see him) and correlating this information with the location contained in their queries [7].

Several countries have taken legal initiative to cope with privacy problems related to electronic communications. In Europe, the European directive on Data Protection and Privacy [8] agrees on a set of measures to assure the privacy of the users of telecommunications technologies such as LBS. Similarly, the Wireless Privacy Protection Act [9] does the same in the US. Unfortunately, all these measures regulate well-established business models but they can hardly be applied to the new LBS that arise in ad-hoc networks created and dismantled on the fly.

Although there are many other relevant topics related to LBS (e.g profile anonymisation [10, 11], trajectories analysis [12, 13], privacy in location-based community services [14], etc.), in this article we concentrate on the methods to protect the location privacy of LBS users who send their location to an LBS provider.

1.1 Contribution and plan of this article

In this article, we provide a survey of the most relevant and recent schemes designed to offer location privacy to LBS users. We analyse, organise and classify them in two main groups: (i) TTP-based schemes and (ii) TTP-free schemes. Moreover, we sketch some ideas on the future of location privacy in LBS and some lines for future research.

The rest of the article is organised as follows. In Section 2 we suggest a classification of the methods for location privacy in LBS proposed in the literature. Section 3 is devoted to the analysis of TTP-based schemes and Section 4 studies TTP-free approaches. Finally, Section 5 contains a brief discussion and some suggestions for future research.

2 Classification of methods for location privacy in LBS

In the simplest form of communication between an LBS user (U) and an LBS provider (P), the former sends a simple query (Q) containing an ID, his location (L) and a request for information (I) that he wants to retrieve from P. Thus, a

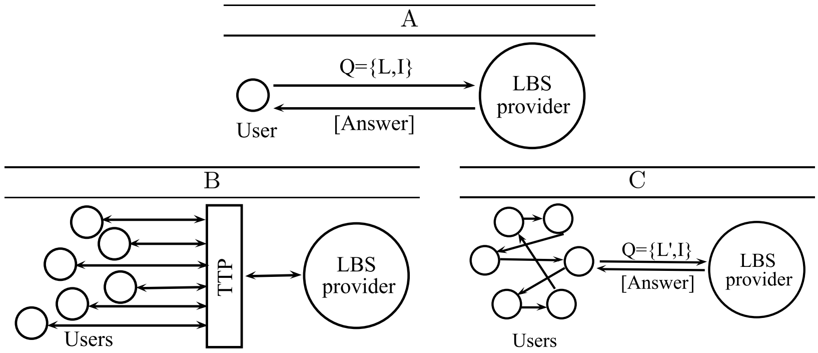


Fig. 1. **A**: Simple communication scheme with an LBS user and an LBS provider. **B**: Communication scheme between an LBS user, an intermediate trusted entity and an LBS provider. **C**: Communication scheme between a set of collaborative LBS users and an untrusted LBS provider. Note that in this scheme location information is not the real one (L), but a perturbed one (L') and no TTP is used.

simple query sent from U to P can be $Q = \{ID_U, L, I\} = \{ID_U, x_U, y_U,$ "Where is the closest bus station?"$\}$ (cf. Figure 1.A). By sending their current locations to P, LBS users assume that P manages their data honestly and refrains from any misuse. However, LBS providers cannot always be trusted and more complex communication schemes are needed.

Most of the solutions proposed in the literature to address the location privacy problem are based on Trusted Third Parties (TTP), i.e. entities which fully guarantee the privacy of their users. Although this approach is widely accepted, it simply moves users' trust from LBS providers to intermediate entities. By doing so, LBS providers are no longer aware of the real locations and identities of the users; trust and, by extension, power are handed over to intermediate entities such as brokers, pseudonymisers or anonymisers. The problem is that users are not necessarily satisfied by completely trusting intermediate entities or providers, especially after the recent scandals related to the disclosure of personal data by this kind of trusted entities[1] (cf. Figure 1.B).

The main difference between the simple communication scheme and the TTP-based one is that in the latter the set of intermediate entities can be expected to be smaller than the number of service providers. Therefore, intermediate entities can be well-known and the risk of trusting a dishonest entity is lessened. However, due to the above mentioned scandals, many users would prefer to trust

[1] In Autumn 2007, several data privacy disasters happened in the UK connected to Her Majesty's Revenue and Customs. Two computer disks full of personal data on 25 million British individuals disappeared; HMRC also lost another disk containing pension records of 15,000 people and a laptop containing personal data on 400 people. In 2006 in the U.S, data on 26.5 million people were stolen from the home of an employee of the Department of Veterans Affairs, and queries by 658,000 users were disclosed by the AOL search engine.

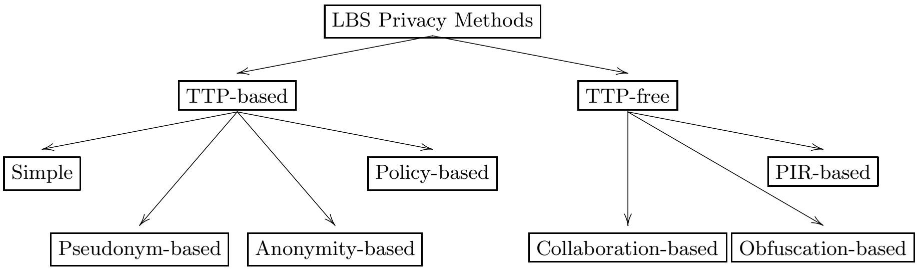


Fig. 2. Classification of location privacy methods for LBS

nobody, which leads to TTP-free schemes. These represent a substantial change of paradigm (cf. Figure 1.C). Instead of trusting a third party, users collaborate to protect their privacy. As it is explained in Section 4, there is not even need to trust the users one collaborates with. Figure 2 depicts our proposed classification of location privacy methods. The main aim of the classification is to emphasise the rigid dichotomy between these two paradigms: (i) TTP-based methods and (ii) TTP-free methods. In the following sections we review some of the most relevant representatives of TTP-based and TTP-free methods.

3 Privacy in TTP-based schemes

TTP-based schemes are very common because they are easy to understand/develop, and because, in general, they offer a reasonable trade-off between efficiency, accuracy and privacy. Moreover, some of the ideas used in these schemes arose in more mature fields like e-commerce.

In the simple scheme described in Section 2, users send their location information and queries directly to the LBS provider. In this scheme, whatever location privacy LBS users can get depends on the honest behaviour of the LBS provider.

In the following sections we concentrate on some TTP-based schemes that aim to protect the location privacy of the users.

3.1 Policy-based schemes

Policy-based schemes are one step forward in LBS privacy with respect to the simple scheme. Although the conceptual framework is the same (i.e. a user submits queries to a provider), in this case, providers adhere to a set of privacy policies known by users. Hence, if providers do not properly follow their privacy policies, users have the right to ask for a compensation and/or take legal action against providers.

Privacy policies are legal notices that contain statements defining what service providers can do with their users' personal data. Privacy policies are published by service providers, and users decide whether such policies are acceptable to them. These policies refer to many concepts and specific languages are used to define them [15, 16]. Users reach an agreement with providers about which data are collected, what are these data used for and how they can be distributed to third parties. In this kind of schemes, privacy is understood as the ability of individuals to decide when, what, and how information about them is disclosed to others. Ideally, users can choose amongst a variety of policies. So, depending on the selected policy, users can save some money but, in return, providers can distribute/sell some of their data.

These schemes are widely used on the Internet by e.g. e-commerce sites which define their privacy policies in e.g. P3P (Platform for Privacy Preferences) [17]. They have been used for automotive telematics [18], and the Geopriv (Geographic Location/Privacy) Charter of the IETF proposes their use for LBS also [19]. A recent study on the use of policies and access control techniques can be found in [20].

3.2 Pseudonymisers

Pseudonymisers are the simplest intermediate entity between LBS users and providers. Pseudonymisers receive queries from users and, prior to forwarding them to LBS providers, they replace the real IDs of the users by fake ones (i.e. pseudonyms). In this way, the real user IDs remain hidden to the provider, but pseudonymisers must store the real IDs and their corresponding pseudonyms in order to forward the answers from the providers to the users. Clearly, users must completely trust pseudonymisers, because the latter see all the location information on the former.

The main problem of this technique is that an attacker (e.g. the LBS provider herself) can infer the real identity of the user by linking the user location with e.g. a public telephone directory (e.g. by using the aforementioned RSI or OI attacks [7]).

3.3 Anonymisers

Anonymisers are the most sophisticated option in TTP-based location privacy. Instead of taking care of policies or users' identifiers, anonymisers assume that communications are anonymous, i.e. LBS providers do not require an ID to answer queries[2]. Anonymisers aim to hide users true identity with respect to emitted location information. In this section we concentrate on techniques that hide the location information of users and we assume that identifier abstraction is already guaranteed.

[2] If this assumption was not made, it would be easy to track a given LBS user by simply checking the ID or the pseudonym (like in the case of pseudonymisers).

A very common way to hide the real location of the users from the LBS provider is by using the k-anonymity property. k-Anonymity is an interesting approach to face the conflict between information loss and disclosure risk, suggested by Samarati and Sweeney [21–24]. Although it was designed for application in databases by the Statistical Disclosure Control (SDC) community, k-anonymity has been adapted to LBS privacy. In this context, we say that the location of a user is k-anonymous if it is indistinguishable from the location of another $k-1$ users. So, the fundamental idea behind k-anonymisers is to replace the real location of the user by cloaking areas in which at least k users are located. Anonymisers transform locations (x, y) at time t to $([x1, x2], [y1, y2], [t1, t2])$ where $([x1, x2], [y1, y2])$ is the rectangular area containing (x, y) between times $t1$ and $t2$ such that $t \in [t1, t2]$. By doing so, LBS providers cannot easily determine which of the k users in the cloaking area is really submitting the query.

Many examples of this kind of approach and other similar ones based on cloaking can be found in the literature [7, 25, 26]. One of the most recent advances in anonymisers is proposed in [27], where an extension of a previous anonymiser version [25] is proposed. The proposed anonymiser allows a user to define his personal privacy requirements, i.e. the number k of users amongst which he wants to be anonymised, and the maximum delay and location perturbation he is willing to accept. The proposal is resilient against identification attacks such as RSI and OI. However, it has some important drawbacks which, as we explain in the next section, can be avoided by TTP-free approaches: (i) the architecture relies on a TTP, so that the user must completely trust the platform mediating between him and the LBS provider; (ii) it is assumed that LBS providers are not malicious but semi-honest, which might turn out to be too much of an idealisation; and (iii) the architecture is centralised, which makes it vulnerable to Denial of Service (DoS) attacks.

In [28] a similar method called PrivacyGrid is described. Although the anonymiser described in [27] and the PrivacyGrid approach are very similar, the latter seems to be more efficient due to the cloaking techniques based on grids (i.e. bottom-up, top-down and hybrid) that it uses. Moreover PrivacyGrid adds the l-diversity property to the already considered k-anonymity one. By doing so, the privacy of LBS users is improved. Although PrivacyGrid seems to improve the proposal in [27], it mainly suffers from the same shortcomings.

Current research on anonymisers focuses on improving the efficiency of the intermediaries and designing highly personalised services able to guarantee the privacy of the users.

4 Privacy in TTP-free schemes

Due to the shortcomings of the TTP-based schemes, other methods that do not rely on TTPs have been proposed. First, we consider the collaboration methods that aim to obtain the same results (e.g. k-anonymity, l-diversity, efficiency) than the ones based on TTP. Then, we pay attention to the methods based on the obfuscation of the real location without collaboration. Finally we point out a new location privacy trend based on Private Information Retrieval (PIR).

4.1 Collaboration-based methods

In [29], the first collaborative TTP-free algorithm for location privacy in LBS is proposed. The user perturbs his location by adding zero-mean Gaussian noise to it. Then the user broadcasts his perturbed location and requests neighbours to return perturbed versions of their locations. Amongst the replies received, the user selects $k-1$ neighbours such that the group formed by the locations of these neighbours and his own perturbed location spans an area A satisfying $A_{min} < A < A_{max}$, where A_{min} is a privacy parameter (the minimum required area for cloaking) and A_{max} is an accuracy parameter (the maximum area acceptable for cloaking). Finally, the user sends to the LBS the centroid of the group of k perturbed locations including his own. Since users only exchange perturbed locations, they do not need to trust each other for privacy. On the other hand, perturbations tend to cancel out each other in the centroid, so accuracy does not degrade[3]. This method does not achieve k-anonymity because the centroid is only used by a single user to identify himself. In addition, due to the noise cancellation, users cannot use this method several times without changing their location. In [30], a similar peer-to-peer scheme for location privacy is presented. Its main idea is to generate cloaking areas as in [29]: users must find other users in their cover range and share their location information. Once this information is known, users can send their queries to LBS providers using the cloaking area instead of their real locations. The main shortcoming of this proposal is that users must trust other users because they exchange their real locations. Thus, a malicious user can easily obtain and publish the location of other users. Although we classify this technique as a TTP-free technique, it can also be understood as a distributed TTP-based scheme, where each user is a TTP.

In [31], the authors propose a method based on Gaussian noise addition to compute a fake location that is shared by k users (unlike in [29]). Thus, all k users use the same fake location and the LBS provider is unable to distinguish one user from the rest, so that their location becomes k-anonymous. This method was extended to support non-centralised communications in [32]. The proposal is based on a stack of modules that progressively increase the privacy achieved by users. The basic module is equivalent to the method described in [30] where users have to trust each other because they share their location. Once they know the locations of other users, they can compute a centroid that they use as their fake location. In order to allow users to exchange their location without trusting other peers, a second module that perturbs the location is added. This module adds Gaussian noise with zero mean to the real location of users. As explained above, the centroid of locations perturbed with zero-mean Gaussian noise is quite similar to the centroid of unperturbed locations. However, if this procedure is

[3] The average of k zero-mean perturbations with variance σ^2 has zero mean and variance σ^2/k.

repeated several times with static users (i.e. users that do not change their location substantially), their real location could be deduced because of the noise cancellation (this is the main problem of [29]). To prevent this, the protocol uses privacy homomorphisms [33] to guarantee that users cannot see the real locations of other users whilst still being able to compute the centroid. Finally, a module that distributes users in a chain is added to avoid denial of service attacks to the central user. At the end of the protocol users become k-anonymous and their location privacy is secured. However, the main problem of this proposal is that it cannot provide a lower bound of the location error.

4.2 Obfuscation-based methods

Obfuscation is a TTP-free alternative to collaboration-based methods. Obfuscation can be understood as the process of degrading the quality of information about a user's location, with the aim to protect that user's privacy [34]. Some methods like the ones described in previous sections (e.g. cloaking methods) can be understood as special kinds of obfuscation because they basically modify the location information in several ways to improve user's privacy. However, we classify them in different categories because they need TTPs and/or achieve other properties such as k-anonymity or l-diversity.

In [35] an obfuscation method based on imprecision is presented. The space is modelled as a graph where vertices are locations and edges indicate adjacency. Hence, in order to obtain an imprecise location, the user sends a set of vertices instead of the single vertex in which he is located. The LBS provider cannot distinguish which of the vertices is the real one. The article proposes negotiation algorithms that allow users to increase the QoS whilst maintaining their privacy. The main problem of this technique is that users and providers must share the graph modelling the space (cf. [36] for a comprehensive approach to imprecision in location systems). Some other recently proposed obfuscation methods can be found in [37], where the real location of LBS users is replaced by circular areas of variable centre and radius.

SpaceTwist [38] is the most recent proposal for non-collaborative TTP-free location privacy. SpaceTwist generates an anchor (i.e. a fake point) that is used to retrieve information on the k nearest points of interest from the LBS provider. After successive queries to the LBS provider, SpaceTwist is able to determine the closest interest point to the real location whilst the LBS provider cannot derive the real location of the user. The main advantages of this method are: (i) no TTP and no collaboration are needed; (ii) the closest interest point is always found; (iii) the location of the user is hidden in a controlled area. However, due to the lack of collaboration, this method is not able to achieve the k-anonymity and/or the l-diversity properties.

4.3 PIR-based methods

A totally different approach to TTP-free LBS privacy is proposed in [39]. In that article, Private Information Retrieval (PIR) is used to provide LBS users with location privacy. Although the idea of using PIR techniques is promising, the proposed approach requires the LBS provider to co-operate with users by following the PIR protocol; this prevents the use of this method in real environments, where LBS providers simply answer queries containing a location or an area without any regard for location privacy. However, if this shortcoming was solved and without significant computation and efficiency penalties, using collaborative PIR amongst peers (i.e. users) could be a really promising future research line.

5 Discussion and future work

In the above sections we have reviewed some of the most recent and relevant contributions to location privacy protection in LBS. There is a clear distinction between TTP-based schemes and the TTP-free ones. Although TTP-based schemes are the most common ones, TTP-free schemes seem superior in terms of privacy due to the following shortcomings of intermediate TTPs: (i) TTPs are critical points which can be attacked; (ii) TTPs are bottlenecks; (iii) There must be many users subscribed to a TTP for the latter to be able to compute suitable cloaking regions (offering sufficient privacy and accuracy).

In general TTP-based schemes are weak because users rely on a single trusted entity. This entity can be impersonated by a bogus TTP created by the attacker, in which case all the information shared by users with the bogus TTP falls in the hands of the attacker. A way to mount such an attack is to tamper with transmitters or use a more powerful signal.

Despite being inferior regarding privacy, TTP-based schemes are easier to implement than collaborative-based methods because all the infrastructure required by users to circumvent the use of a TTP is not necessary. However, obfuscation-based methods are also easy to implement. We believe that there is room in the market for both approaches.

The use of k-anonymity and l-diversity properties must be carefully considered because in some scenarios they are insufficient to preserve user's privacy [40]. In our opinion, there are a lot of opportunities for synergy between future work in PIR and TTP-free LBS privacy. Indeed, current PIR techniques face the (very serious) limitation of needing co-operation from the database server in following the PIR protocol. If practical PIR protocols are developed which do not need such a co-operation, it will be possible to use them for TTP-free location privacy: if a query can be submitted to a non-co-operative commercial LBS server in such a way that the latter does not learn what the query is about (i.e. the location supplied by the user), then one obtains a TTP-free LBS privacy protocol.

Acknowledgements

The authors are solely responsible for the views expressed in this article, which do not necessarily reflect the position of UNESCO nor commit that organisation. This work was partly supported by the Spanish Ministry of Education through projects TSI2007-65406-C03-01“E-AEGIS” and CONSOLIDER INGENIO 2010 CSD2007-0004 “ARES”, and by the Government of Catalonia under grant 2005 SGR 00446.

The first author thanks Dr. Paul A. Karger (IBM T. J. Watson Research Center) for his help in providing relevant references and material.

References

1. Drane, C., Macnaughtan, M., Scott, C.: Positioning GSM telephones. IEEE Communications Magazine **36**(4) (April 1998) 46 – 54 , 59
2. Simcock, T., Hillenbrand, S.P., Thomas, B.H.: Developing a location based tourist guide application. In Johnson, C., Montague, P., Steketee, C., eds.: ACSW Frontiers ’03: Proceedings of the Australasian information security workshop conference on ACSW frontiers 2003. Volume 21 of CRPIT., Darlinghurst, Australia, Australian Computer Society, Inc. (February 2003) 177–183
3. Yoo, K., Park, D., Rhee, B.: Development of a location-based dynamic route guidance system of korea highway corporation. In Satoh, K., ed.: Proceedings of the Eastern Asia Society for Transportation Studies. Volume 5., Bangkok, Eastern Asia Society for Transportation Studies (2005) 1449 – 1463
4. Reed, J.H., Krizman, K.J., Woerner, B.D., Rappaport, T.S.: An overview of the challenges and progress in meeting the E-911 requirement for location privacy. IEEE Communications Magazine **36**(4) (April 1998) 30 – 37
5. Kölmel, B., Alexakis, S.: Location based advertising. In: The First International Conference on Mobile Business, Athens, Greece (2002)
6. Karger, P.A., Frankel, Y.: Security and privacy threats to ITS. In: The Second World Congress on Intelligent Transport Systems. Volume 5., Yokohama, Japan. (November 1995) 2452 – 2458
7. Gruteser, M., Grunwald, D.: Anonymous usage of location-based services through spatial and temporal cloaking. In: Proceesings of MobiSys 2003: The First International Conference on Mobile Systems, Applications, and Services., San Francisco, CA, USA, USENIX Association, ACM, Sigmobile, ACM (May 2003) 31 – 42
8. The European Parliament and the Council: Directive 2002/58/EC on privacy and electronic communications. Official Journal of the European Communities **201** (July 2002) 37 – 47
9. 108th Congress: H.R. 71: The wireless privacy protection act. In: United States House of Representatives. (2003-4) `http://frwebgate.access.gpo.gov/cgi-bin/getdoc.cgi?dbname=108_cong_bills&docid=f:h71ih.txt.pdf`.
10. Atluri, V., Shin, H.: Efficient security policy enforcement in a location based service environment. In Baker, S., Ahn, G., eds.: Data and Applications Security. Volume 4602 of LNCS., IFIP, Springer Berlin / Heidelberg (2007) 61–76
11. Shin, H., Atluri, V., Vaidya, J.: A profile anonymization model for privacy in a personalized location based service environment. In: 9th International Conference on Mobile Data Management. MDM’08. (2008) 73–80
12. Hoh, B., Gruteser, M.: Protecting location privacy through path confusion. In: Proceedings of the First International Conceference on Security and Privacy for Emerging Areas in Communications Networks. (2005) 194–205
13. Terrovitis, M., Mamoulis, N.: Privacy preservation in the publication of trajectories. In: 9th International Conference on Mobile Data Management. MDM’08. (2008) 65–72

14. Ruppel, P., Treu, G., Küpper, A., Linnhoff-Popien, C.: Anonymous user tracking for location-based community services. In Hazas, M., Krumm, J., Strang, T., eds.: Second International Workshop on Location- and Context-Awareness. Volume 3987 of LNCS., Springer Berlin / Heidelberg (2006) 116 – 133
15. Snekkenes, E.: Concepts for personal location privacy policies. In: ACM Conference on Electronic Commerce. (2001) 48–57
16. Cranor, L.F.: P3P: Making privacy policies more useful. IEEE Security & Privacy **1**(6) (2003) 50–55
17. W3C: Platform for privacy preferences (P3P) project. Webpage (October 2007) `http://www.w3.org/P3P/`.
18. Duri, S., Gruteser, M., Liu, X., Moskowitz, P., Perez, R., Singh, M., Tang, J.M.: Framework for security and privacy in automotive telematics. In: Proceedings of the 2nd international workshop on Mobile commerce, ACM Press New York, NY, USA (2002) 25 – 32
19. Schulzrinne, H., Tschofenig, H., Morris, J., Cuellar, J., Polk, J.: Geolocation policy. Technical report, Internet Engineering Task Force (June 2008) `http://www.ietf.org/internet-drafts/draft-ietf-geopriv-policy-17.txt`.
20. Bauer, L., Cranor, L.F., Reeder, R.W., Reiter, M.K., Vaniea, K.: A user study of policy creation in a flexible access-control system. In Czerwinski, M., Lund, A.M., Tan, D.S., eds.: Proceedings of the 2008 Conference on Human Factors in Computing Systems, ACM (2008) 543–552
21. Samarati, P.: Protecting respondents' identities in microdata release. IEEE Transactions on Knowledge and Data Engineering **13**(6) (2001) 1010–1027
22. Samarati, P., Sweeney, L.: Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression. Technical report, SRI International (1998)
23. Sweeney, L.: Achieving k-anonymity privacy protection using generalization and suppression. International Journal of Uncertainty, Fuzziness and Knowledge Based Systems **10**(5) (2002) 571–588
24. Sweeney, L.: k-anonimity: A model for protecting privacy. International Journal of Uncertainty, Fuzziness and Knowledge Based Systems **10**(5) (2002) 557–570
25. Gedik, B., Liu, L.: A customizable k-anonymity model for protecting location privacy. In: Proceedings of the IEEE International conference on Distributed Computing Systems (ICDS'05). (2005) 620 – 629
26. Cheng, R., Zhang, Y., Bertino, E., Prabhakar, S.: Preserving user location privacy in mobile data management infrastructures. In: 6th Workshop on Privacy Enhancing Technologies (PET). Volume 4258 of Lecture Notes in Computer Science., Springer Berlin / Heidelberg (2006) 393 – 412
27. Gedik, B., Liu, L.: Protecting location privacy with personalized k-anonymity: Architecture and algorithms. IEEE Transactions on Mobile Computing **7**(1) (January 2008) 1 – 18
28. Bamba, B., Liu, L., Pesti, P., Wang, T.: Supporting anonymous location queries in mobile environments with privacygrid. In: International World Wide Web Conference WWW. (2008) 237–246
29. Domingo-Ferrer, J.: Microaggregation for database and location privacy. In Etzion, O., Kuflik, T., Motro, A., eds.: Next Generation Information Technologies and Systems-NGITS. Volume 4032 of LNCS., Springer Berlin / Heidelberg (2006) 106–116
30. Chow, C., Mokbel, M.F., Liu, X.: A peer-to-peer spatial cloaking algorithm for anonymous location-based services. In: GIS '06: Proceedings of the 14th annual

ACM international symposium on Advances in geographic information systems, Arlington, Virginia, USA, ACM (November 2006) 171–178

31. Solanas, A., Martínez-Ballesté, A.: Privacy protection in location-based services through a public-key privacy homomorphism. In: Fourth European PKI Workshop: theory and practice. Lecture Notes in Computer Science, Springer Berlin / Heidelberg (2007) 362 – 368 Palma de Mallorca, Spain.
32. Solanas, A., Martínez-Ballesté, A.: A TTP-free protocol for location privacy in location-based services. Computer Communications **31**(6) (April 2008) 1181–1191
33. Okamoto, T., Uchiyama, S.: A new public-key cryptosystem as secure as factoring. In: Advances in Cryptology EUROCRYPT'98. Volume 1403 of Lecture Notes in Computer Science., Springer Berlin / Heidelberg (1998) 308
34. Duckham, M., Kulit, L.: Location Privacy and Location-Aware Computing. Number 3. In: Dynamic and Mobile GIS: Investigating Changes in Space and Time. CRC Press (2007) 35–52
35. Duckham, M., Kulit, L.: A formal model of obfuscation and negotiation for location privacy. In: Pervasive Computing. Volume 3468 of LNCS., Springer Berlin / Heidelberg (2005) 152–170
36. Duckham, M., Mason, K., Stell, J., Worboys, M.: A formal approach to imperfection in geographic information. Computers, Environment and Urban Systems **25**(1) (2001) 89–103
37. Ardagna, C.A., Cremonini, M., Damiani, E., S. De Capitani di Vimercati, Samarati, P.: Location privacy protection through obfuscation-based techniques. In Baker, S., Ahn, G., eds.: Data and Applications Security. Volume 4602 of LNCS., IFIP (2007) 47 – 60
38. Yiu, M.L., Jensen, C.S., Huang, X., Lu, H.: Spacetwist: Managing the trade-offs among location privacy, query performance, and query accuracy in mobile services. In: IEEE 24th International Conference on Data Engineering ICDE'08. (2008) 366–375
39. Ghinita, G., Kalnis, P., Khoshgozaran, A., Shahabi, C., Tan, K.: Private queries in location based services: Anonymizers are not necessary. In: SIGMOD '08: Proceedings of the 2008 ACM SIGMOD international conference on Management of data, Vancouver, BC, Canada, ACM (June 2008) 121 – 132
40. Pareschi, L., Riboni, D., Bettini, C.: Protecting users' anonymity in pervasive computing environments. In: Sixth Annual IEEE International Conference on Pervasive Computing and Communication (PERCOM'08), IEEE Computer Society (2008) 11–19

Privacy in Georeferenced Context-aware Services: A Survey

Daniele Riboni, Linda Pareschi, and Claudio Bettini

EveryWare Lab, D.I.Co., University of Milano
via Comelico 39, I-20135 Milano, Italy
{riboni,pareschi,bettini}@dico.unimi.it

Abstract. Location based services (LBS) are a specific instance of a broader class of Internet services that are predicted to become popular in a near future: context-aware services. The privacy concerns that LBS have raised are likely to become even more serious when several context data, other than location and time, are sent to service providers as part of an Internet request. This paper provides a classification and a brief survey of the privacy preservation techniques that have been proposed for this type of services. After identifying the benefits and shortcomings of each class of techniques, the paper proposes a combined approach to achieve a more comprehensive solution for privacy preservation in georeferenced context-aware services.

1 Introduction

It is widely recognized that the success of context-aware services is conditioned to the availability of effective privacy protection mechanisms (see, e.g., [1, 2]). Techniques for privacy protection have been thoroughly studied in the field of databases, in order to protect microdata released from large repositories. Recently some of these techniques have been extended and integrated with new ones to preserve the privacy of users of Location Based Services (LBS) against possibly untrusted service providers as well as against other types of adversaries [3]. The domain of service provisioning based on location and time of request introduces novel challenges with respect to traditional privacy protection in microdata release. This is mainly due to the dynamic nature of the service paradigm which requires a form of *online* privacy preservation technique as opposed to an *offline* one used, for example, in the publication of a view from a database. In the case of LBS, specific techniques are also necessary to process the spatio-temporal information describing location and time of request which is also very dynamic. On the other hand, location and time are only two of the possibly many parameters characterizing the context of an Internet service request. Indeed, context information goes far beyond location and time, including data such as personal preferences and interests, current activity, physiological and emotional status, and data collected from body-worn or environmental sensors, just to name a few. Privacy protection techniques specifically developed for LBS are often insufficient and/or inadequate when applied to generic context-aware services.

Consider, for instance, cryptographic techniques proposed for LBS (e.g., [4, 5]). These techniques provide strong privacy guarantees at the cost of high computational overhead on both the client and server side; moreover, they introduce expensive communication costs. Hence, while they may be profitably applied to simple LBS such as nearest neighbor services, it is unlikely that they would be practical for complex context-aware services. On the other hand, obfuscation techniques proposed for LBS (e.g., [6, 7]) are specifically addressed to location information; hence, those techniques cannot be straightforwardly applied to other contextual domains. With respect to techniques based on identity anonymity in LBS (e.g., [8, 9]) we point out that, since many other kinds of context data besides location may help an adversary in identifying the owner of those data, the amount of context data to be generalized in order to enforce anonymity is large. Hence, even if filtering techniques can be used for improving the service response, it could happen that in order to achieve the desired anonymity level, context data become too general to provide the service at an acceptable quality level [10]. For this reason, specific anonymity techniques for generic context-aware services are needed.

Moreover, in pervasive computing environments context-aware services can exploit data provided by sensors deployed in the environment that can constantly monitor context data. Hence, if those context sources are compromised, an adversary's inference abilities may increase taking advantage of the observation of users' behavior and of up-to-date context information. Defense techniques for privacy preservation proposed for LBS do not consider this kind of inference capabilities, since location and time are the only contextual parameters that are taken into account. As a result, protecting against the above mentioned kind of attacks requires new techniques.

In this paper we survey privacy protection techniques for georeferenced context-aware services. As depicted in Figure 1, the general privacy threat we are facing is the release of *sensitive associations* between a user's identity and the information that she considers private. The actual privacy risk certainly depends on the adversary's model; for the purpose of this survey, unless we mention specific attacks, we adopt the general assumption that an adversary may obtain service requests and responses as well as publicly available information.

We distinguish different types of defense techniques that can be used to contrast the privacy threat.

- **Network and cryptographic protocols.** These are mainly used to avoid that an adversary can access the content of a request or response while it is transmitted as well as to avoid that a network address identifies the location and/or the issuer of a request.
- **Access control mechanisms.** These are used to discriminate (possibly based on context itself) the entites that can obtain certain context information.

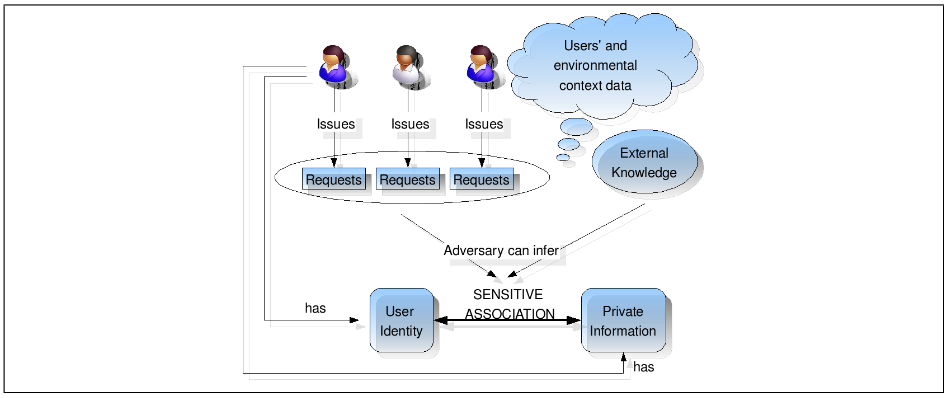


Fig. 1. The privacy threat

- **Obfuscation techniques.** Under this name we group the techniques, usually based on generalization or partial suppression, that limit the disclosure of private information contained in a request. Intuitively, they control the release of the second part of the association describing the privacy threat.
- **Identity anonymization techniques.** These are techniques that aim at avoiding the release of the first part of the association, i.e., the identity of the issuer. The goal is to make the issuer indistinguishable among a sufficiently large number of individuals.

This classification may apply as well to defenses against LBS privacy threats, however our description of available approaches and solutions will be focused on those for more complex context-aware services. Sections 2, 3, 4, and 5 address each of the above types of defenses, respectively. Based on the weaknesses emerged from the analysis of the existing techniques, in Section 6 we advocate the use of a combined approach, present preliminary proposals, and illustrate the general characteristics that a comprehensive combined approach may have. Section 7 concludes the paper.

2 Network and cryptographic protocols

The development of context-aware services received impulse by technological progresses in the area of wireless communications, mobile devices, and sensors. The use of wireless channels, and more generally insecure channels, poses a first threat for the users' privacy since it makes easier for an adversary to acquire service requests and responses by eavesdropping the communication or analyzing traffic on the network. In the literature, several models have been proposed for privacy preservation in context-aware systems. While some of them rely on a *centralized* architecture with a single trusted entity in charge of ensuring the users' privacy, other models rely on a *decentralized* architecture in which mobile devices use direct communication channels with service providers. *decentralized* architectures in which mobile communication channels with service providers. In both cases, two natural countermeasures for privacy attacks are: a) implement secure communication channels so that no third party can obtain requests/responses while they are in transit, and b) avoid the recognition of the client's network address, even by the service provider, which may be untrusted.

In order to protect point-to-point communications, in addition to standard wireless security, different cryptographic techniques can be applied. One possibility is clearly for applications to rely on SSL to encrypt communication; an alternative (or additional) possibility is to provide authentication, authorization and channel encryption through systems like Kerberos ([11]). Kerberos is based on a centralized entity, *Key Distribution Center* (KDC), in charge of authenticating clients and servers in the network, and providing them with the keys needed for encrypting the communications. The centralized model that inspires Kerberos does not protect from attacks aimed at acquiring the control of the KDC entity. Specific solutions to communication protection also depend on the considered architecture and adversary's model, and are outside the scope of this paper.

Different approaches ([12, 13]) aim at guaranteeing a certain degree of anonymity working at the IP level. The Tarzan system ([12]) adopted a solution based on a network overlay that clusters nodes in subnetworks called *domains* on the base of their IP addresses. The IP hiding is achieved by the substitution of the sender's IP with the pseudonym corresponding to its domain. Moreover, when a node needs to send a packet, its communications are filtered by a special server called *mimic* that is in charge of *i)* substituting the IP and other information that could reveal the sender identity with the adequate pseudonym, and *ii)* of setting a virtual path (*tunnel*) that guarantees the communication encryption.

Most solutions presented in the literature apply a combination of routing protocols for IP hiding, and cryptographic techniques ([14]) to protect from eavesdropping over the communication channel. Onion Routing ([15]) implements both the features of IP hiding and message encryption. In order to preserve the sender's IP address, each message travels towards the receiver via a series of proxies, called *onion routers*, which choose the next component of the path setting an unpredictable route. Each router in the path re-encrypts the message before forwarding it to the next router. However, even these solutions suffer from attacks aimed at acquiring the control of one or more nodes of the network.

A different application of a privacy-preserving routing protocol is presented in [16]: the proposed solution has been designed for protecting the user's privacy while moving in smart environments. This solution is based on a hierarchy of trusted servers where the leaves, called *portals*, are aware of the user's location, while internal nodes are aware of services provided by the environment. The user accesses the network through a portal and, according to her privacy preferences, she is assigned to an internal node, called *lighthouse*, that has the task of filtering and encrypting all the communications between the user and the service provider. The lighthouse does not know the user's position but is aware of the next hop in the server hierarchy composing the path to the user's portal. Similarly, the portal does not know which service the user is asking for, but it is aware of the path to the chosen lighthouse. The privacy preservation is achieved decoupling position data from both the identity information and other context parameters. However, this approach requires the servers in the hierarchy to be trusted and it does not protect by privacy attacks performed acquiring the control of one of the nodes in the structure.

The use of cryptographic techniques can also be extended to hide from the service provider the exact request parameters as well as the response. This approach has been proposed in the area of LBS where location information is often considered sensitive by users. In particular, solutions based on this approach aim at retrieving the nearest neighbor (NN) point of interest (*poi*) with respect to the user position at the time of the request.

A first solution was proposed in [4]: the authors propose a form of encrypted query processing combining the use of a data structure suited for managing spatial information with a cryptographic schema for the secret sharing. On the server side, location data are handled through a *directed acyclic graph* (*DAG*), whose nodes correspond to Voronoi regions obtained by a tessellation of the space with respect to *poi*s stored by the service provider. The query processing is performed according to the protocol proposed in [17] that allows a client to retrieve the correct Voronoi area without communicating its precise location. The drawback of this solution is that, in order to resolve a NN query, the user needs to send a number of queries that is proportional to the depth of the *DAG* instead of a single request. The consequent communication overhead impacts on the network traffic and on the response time, which are commonly considered important factors in mobile computing.

Recently, a cryptographic approach inspired by the Private Information Retrieval (PIR) field was proposed in [5]. The service provider builds a Voronoi tessellation according to the stored *poi*s, and superimposes on its top a regular grid of arbitrary granularity. In order to obtain the response to a NN query the privacy preservation mechanism relies on a PIR technique that is used for encrypting the user query, and for retrieving part of the location database without revealing spatial information. Some of the strong points of this solution are that location data are never disclosed; the user's identity is confused among identities of all users; and no trusted third party is needed to protect the users' privacy. However, since mobile devices are often characterized by limited computational capability, the query encryption and the answer processing performed at the client side have a strong impact on service response time, network and power consumption. In particular, when applied to context-aware services that perform the adaptation on a wide set of heterogeneous context data, this technique may result in unacceptable computation overhead both at the client and at the server side.

3 Access control in context-aware systems

Pervasive computing environments claim for techniques to control release of data and access to resources on the basis of the context of users, environment, and hardware/software entities. In general, the problem of access control [18] consists in deciding whether to authorize or not a requesting entity (*subject*) to perform a given *action* on a given resource (*object*). Access control mechanisms have been thoroughly studied in many fields, including operating systems, databases, and distributed systems. However, the characteristic features of pervasive environments introduce novel issues that must be taken into account for devising effective access control mechanisms. In particular, differently from centralized organizational domains, pervasive environments are characterized by the intrinsic decentralization of authorization decisions, since the object owners (users, services, infrastructures) are spread through the environment, and may adopt different policies regarding disclosure of private information. Hence, specific techniques to deal with the mobility and continuously changing context of the involved entities are needed to adapt authorizations to the current situation.

To this aim various techniques for context-aware access control have been recently proposed. Context-aware access control strategies fall in two main categories. The first category is the one of techniques aimed at granting or denying access to resources considering the context of the requesting user and of the resource (see, e.g., [19–21]). The second category is the one of techniques aimed at controlling the release of user's context data on the basis of the context of the requesting entity and of the user herself. In this section we concentrate on techniques belonging to the latter category. On the contrary, techniques belonging to the former category are outside the scope of this paper, and will not be reviewed; however, we point out that, since those techniques imply the release of users' context data to the access control mechanism, generally they also adopt strategies to enforce users' privacy policies.

Proposed context-aware access control mechanisms can be roughly classified in those that derive from *discretionary (DAC)* [22] and those that derive from *role-based (RBAC)* [23] access control. In DAC systems, the owner of each object is in charge of stating policies to determine the access privileges on the basis of the subject identity. These techniques are well suited to domains in which subjects do not belong to a structured organization (e.g., they are well suited to generic Internet services), since they are released from the burden of managing groups or roles of subjects. On the other hand, techniques based on RBAC (in which the access privileges depend on the subject role) are well suited to structured organization domains (like, e.g., hospitals, companies), since the definition of functional roles simplifies the management of access control policies.

Other techniques related to access-control in context-aware systems include the use of access-rights graphs and hidden constraints (e.g., [24]) as well as *zero-knowledge proof theory* [25] (e.g., [26]). These are called *secret authorization* mechanisms, since they allow an entity to certify to a verifier the possession of private information (e.g., context data) revealing neither the authorization policies nor the secret data.

In the following we briefly describe the access control techniques for context-awareness derived from DAC and RBAC models, respectively.

Techniques derived from DAC. Even early approaches to discretionary access control allowed the expression of conditions to constrain permissions on the basis of the spatial and temporal characterization of the subject. For instance, in a bank setting, access to customer accounts could be acknowledged to authorized personnel only during working hours and from machines located within the bank. More recently, access control techniques specifically addressed to the protection of location information (e.g., [27]) have been proposed. However, the richness and dynamics of contextual situations that may occur in pervasive and mobile computing environments claim for the definition of formal languages to express complex conditions on a multitude of context data, as well as sufficiently expressive languages to represent the context itself. To this aim, *Houdini* [28] provides a comprehensive formal framework to represent dynamic context data, integrate them from heterogeneous sources, and share context information on the basis of users' privacy policies. In particular, privacy policies can be expressed considering the context of the data owner (i.e., the user) and the context of the subject. As an example, a user of a service for locating friends could state a policy to disclose her current location to her friends only if her mood is *good* and her current activity is not *working*. Privacy policies in *Houdini* are expressed in a restricted logic programming language supporting rule chaining but no cycles. Rules preconditions express conditions on context data, while postconditions express permissions to access contextual information; reasoning with the resulting language has low computational complexity. Policy conflict resolution is based on explicit rule priorities.

Another relevant proposal, specifically addressed to the preservation of mobile customers privacy, can be found in [29]. That work proposes an access control system aimed at controlling the release of private data based on time, location, and customer's preferences. For instance, a user could state a policy to disclose her location and profile information only during the weekend and if she is in a mall, and only in exchange for a discount coupon on items in her shopping list. The proposed solution is based on an intermediary infrastructure in charge of managing location and profiles of mobile users and to enforce their privacy policies. A specific index structure as well as algorithms are presented to efficiently enforce the proposed techniques.

Techniques derived from RBAC. Many other existing approaches to context-aware access control are based on an extension of the RBAC model. As anticipated before, RBAC systems are well-suited to structured organization domains. However, the baseline RBAC model is not adequate to pervasive and mobile computing domains, which are characterized by the dynamics of situations that may determine the role played by a given entity in a given context. For this reason, various proposals have been made to extend RBAC policies with contextual conditions (see, e.g., [19]), and in particular with spatio-temporal constraints (e.g., [30]). More recently, this approach has been applied to the privacy protection of personal context data. A proposal in this sense is provided by the *UbiCOSM* middleware [31], which tackles the comprehensive issue with mechanisms to secure the access not only to services provided by ubiquitous infrastructures, but also to users' context data, based on contextual conditions and roles. The context model of UbiCOSM distinguishes between the *physical* dimension, which describes the spatial characterization of the user, and the *logical* dimension, which describes other data such as the user's current activity and device capabilities. For instance, the context *TouristAtMuseum* is composed by the physical context *AtMuseum* (characterized by the presence of the user within the physical boundaries of a museum) and by the logical context *Tourist* (which defines the user's role as the one of a tourist). Users can declare a policy to control the release of a personal context data as the association between a permission and a context in which the permission applies. Simple context descriptions can be composed in more complex ones by means of logical operators, and may involve the situation of multiple entities. For instance, in order to find other tourists that share her same interests, a user could state a policy to disclose her cultural preferences to a person only if their current context is *TouristAtMuseum* and they are both co-located with a person that is a friend of them both.

Another worth-mentioning system is *CoPS* [32], which provides fine-grained mechanisms to control the release of personal context data, as well as techniques to identify misuse of the provided information. In particular, policies in CoPS are organized in a hierarchical manner, on the basis of the priority level of the policy (i.e., organization-level, user-level, default). Permissions depend on the context and the role of the subject. CoPS supports both administrator and user-defined roles. While the former reflect the hierarchical structure of the organization, the latter can be used to categorize entities in groups, in order to simplify the policy management by users. The system adopts a conflict resolution mechanism based on priorities and on the specificity of access control rules. Moreover, a trigger mechanism can be set up to control the release of particular context data against the frequency of the updates; this technique can be used, for instance, to notify the user in the case someone tries to track her movements by continuously polling her location.

Open issues and remarks. As emerged from the above analysis of the state-of-the-art, the main strong point of techniques derived from DAC consists in the efficiency of the reasoning procedures they employ to evaluate at run-time the access privileges of the requesting entity. This characteristic makes them very well suited to application domains characterized by strict real-time requirements, like telecommunication and Internet services. On the other hand, the roles abstraction adopted by techniques derived from RBAC can be profitably exploited not only in structured organizational domains but also in open environments (like ambient intelligence systems), since heterogeneous entities can be automatically mapped to predefined roles on the basis of the contextual situation to determine their access privileges.

Nevertheless, some open issues about context-aware access control systems are worth to be considered. In particular, like in generic access control systems, a formal model to represent policies and automatically recognize inconsistencies (especially in systems supporting the definition of negative authorizations) is needed; however, only part of the techniques proposed for context-aware computing face this issue. This problem is further complicated by the fact that the privacy policy of a subject may conflict with the privacy policy of an object owner. Proposed solutions for this issue include the use of techniques for secret authorization, like proposed in [24]. Moreover, an evident weakness of these systems consists in their rigidity: if strictly applied, an access control policy either grants or denies access to a given object. This weakness is alleviated by the use of obfuscation techniques (reported in Section 4) to disclose the required data at different levels of accuracy on the basis of the current situation.

A further critical issue for context-aware access control systems consists in devising techniques to support end users in self-defining privacy policies. Indeed, manual policy definition by users is an error-prone and tedious task. For this reason, straightforward techniques to support users' policy definition consists in making use of user friendly interfaces and default policies, like in Houdini and in CoPS, respectively. However, a more sophisticated strategy to address this problem consists in the adoption of statistical techniques to automatically learn privacy policies on the basis of the past decisions of the user. To this aim, [33] propose the application of rough set theory to extract access control policies based on the observation of the user's interaction with context-aware applications during a training period.

As a final remark, we point out that context-aware access control systems do not protect privacy in the case the access to a service is considered a private information by itself (e.g., because it reveals particular interests or habits about the user). To address this issue, techniques aimed at enforcing anonymity exist and are reviewed in Section 5.

4 Obfuscation of context data

In some cases, the strict application of access control mechanisms (i.e., either deny or allow access to a given context data in a given situation) may be a too rigid strategy. For instance, consider the user of a service that redirects incoming calls and messages on the basis of the current activity. Suppose that the service is not completely trusted by the user; hence, since she considers her current activity (e.g., *MeetingCustomers*) a sensitive information, whether to allow or deny the access to her precise current activity may be unsatisfactory. Indeed, denying access to that data would determine the impossibility to take advantage of that service, while allowing access could result in a privacy violation. In this case, a more flexible solution is to *obfuscate* [34] the private data before communicating it to the service provider in order to decrease the sensitivity level of the data. For instance, the precise current activity *MeetingCustomers* could be obfuscated to the more generic activity *BusinessMeeting*. This solution is based on the intuition that each private data is associated to a given sensitivity level, which depends on the precision of the data itself; generally, the lesser the data is precise, the lesser it is sensitive. Obfuscation techniques have been applied to the protection of microdata released from databases (e.g., [35]).

Several techniques based on obfuscation have also been proposed to preserve the privacy of users of context-aware services. These techniques are generally coupled with an access control mechanism to tailor the obfuscation level to be enforced according to the trustiness of the subject and to the contextual situation. However, in this section we concentrate on works that specifically address context data obfuscation. The main research issue in this field is to devise techniques to provide adequate privacy preservation while retaining the usefulness of the data to context-awareness purposes. We point out that, differently from techniques based on anonymity (reviewed in Section 5), techniques considered in this section do not protect against the disclosure of the user's identity.

Various obfuscation-based techniques to control the release of location information have been recently proposed (see, e.g., [36, 6, 7]), based on generalization or perturbation of the precise user's position. One of the first attempts to support privacy in generic context aware systems through obfuscation mechanisms is *semantic eWallet* [37], an architecture to support context-awareness by means of techniques to retrieve users' context data while enforcing their privacy preferences. Users of the semantic eWallet may express their preferences about the accuracy level of their context data based on the requester's identity and on the context of the request. That system supports both *abstraction* and *falsification* of context information. By abstraction, the user can decide to generalize the provided data, or to omit some details about it. For instance, a user involved in a *BusinessMeeting* could decide to disclose her precise activity to a colleague only during working hours and if they both are located within a company building; activity should be generalized to *Meeting* in the other cases. On the other hand, by falsification the user can decide to deliberately provide false information in order to mask her precise current context in certain situations. For instance, a CEO could reveal to her secretary that she is currently *AtTheDentist*, while telling to the other employees that she is involved in a *BusinessMeeting*. In the semantic eWallet, context data are represented by means of ontologies. Obfuscation preferences are encoded as rules whose preconditions include a precise context data and conditions for obfuscation, and postconditions express the obfuscated context data to be disclosed if the preconditions hold.

While in the semantic eWallet the mapping between precise and obfuscated information must be explicitly stated case-by-case, a more scalable approach to the definition of obfuscation preferences is proposed in [38]. That work copes with the multi-party ownership of context information in pervasive environments by proposing a framework to retrieve context information and distributing it on the basis of the obfuscation preferences stated by the data owner. It is worth to note that in the proposed framework the owner of the data is not necessarily the actual proprietary of the context source; instead, the data owner is the person whom the data refers to. For instance, the owner of data provided by a server-side positioning system is the user, not the manager of the positioning infrastructure; hence, the definition of obfuscation preferences about personal location is left to the user. Obfuscation preferences are expressed by conditions on the current context, by specific context data, and by a maximum detail level at which that data can be disclosed in that context. The level of detail of a context data refers to the specificity of that data according to a predefined *obfuscation ontology*. Context data in an obfuscation ontology are organized as nodes into a hierarchy, such that parent nodes represent more general concepts with respect to their children; e.g., the activity *MeetingCustomers* has parent activity *BusinessMeeting*, which in turn has parent activity *Working*. For instance, an obfuscation preference could state to disclose the user's current activity with a *level 2* specificity in the case the requester is *Bob* and the request is made during *working hours*. In the case those conditions hold, the released data is calculated by generalizing the exact current activity up to the second level of the *Activity* obfuscation ontology (i.e., up to the level of the grandchildren of the root node), or to a lower level if the available information is less specific than that stated by the preference. Since manually organizing context data in an obfuscation ontology could be unpractical, a technique to automatically discover reasoning modules able to derive the data at the required specificity level is also presented.

Based on the consideration that the quality of a context information (*QoC*, intended as its closeness to the physical reality it describes) is a strong indicator of privacy sensitiveness, Sheikh et al. propose the use of QoC to enforce users' privacy preferences [39]. In that work, the actual quality of the disclosed context data is negotiated between service providers and users. When a service provider needs a data regarding a user's context, it specifies the QoC that it needs for that data in order to provide the service. On the other hand, the user specifies the maximum QoC she is willing to disclose for that data in order to take advantage of the service. Service requirements and user's privacy preferences are communicated to a middleware that is in charge of verifying if they are incompatible (i.e., if the service requires a data to a quality the user is not willing to provide). If this is not the case, obfuscation mechanisms are applied on that data in order to reach the quality level required by the service provider. QoC is specified on the basis of five indicators, i.e., precision, freshness, spatial and temporal resolution, and probability of correctness. Each context data is associated with five numerical values that express the quality of the data with respect to each of the five indicators. Given a particular context situation, a user can specify her privacy preferences for a context data by defining the maximum quality level for each of the five indicators that she is willing to disclose in that situation. For instance, the user of a remote health monitoring service could state to disclose vague context information to the caregivers when in a non-emergency context, while providing accurate data in the case of emergency.

One inherent weakness of obfuscation techniques for privacy in context-awareness is evident: if the service provider requires a context data to a quality that the user is not willing to disclose, access to that service is not possible. In order to overcome this issue, anonymization techniques (presented in Section 5) have been proposed, which protect from the disclosure of the user's identity, while possibly providing accurate context information.

5 Identity anonymization techniques

While obfuscation techniques aim at protecting the right-hand side of the sensitive association (SA) (see Figure 1), the goal of techniques for identity anonymization is to protect the left-hand side of the SA in order to avoid that an adversary re-identifies the issuer of a request.

In the area of database systems, the notion of k-anonymity has been introduced [40] to formally define when, upon release of a certain database view containing records about individuals, for any specific sensitive set of data in the view, the corresponding individual can be considered indistinguishable among at least k individuals. In order to enforce anonymity it is necessary to determine which attributes in a table play the role of *quasi-identifiers* (*qi*), i.e., data that joined with external knowledge may help the adversary to restrict the set of candidate individuals. Techniques for database anonymization adopt generalization of *qi* values and/or suppression of records in order to guarantee that the set of released records can be partitioned in groups of at least k records having the same value for *qi* attributes (called *qi-groups*). Since each individual is assumed to be the respondent of a single record, this implies that there are at least k candidate respondents for each released record.

The idea of k-anonymity has also been applied to define a privacy metric in location based services, as a specific kind of context-aware services [8]. In this case, the information being released is considered the information in the service request. In particular, the information about the user's location may be used by an adversary to re-identify the issuer of the request if the adversary has access to external information about users' location. Attacks and defense techniques in this context have been investigated in several papers, among which [8, 9]. Moreover, a formal framework for the categorization of defense techniques with respect to the adversary's knowledge assumptions has been proposed in [3]. According to that categorization, when the adversary performs his attack using information contained in a single request the attack is said to be *single-issuer*; otherwise, when the adversary may compare information included in requests by multiple users, the attack is said to be *multiple-issuers*. Moreover, cases in which the adversary can acquire information only during a single time granule are called static (or *snapshot*), while contexts in which the adversary may observe multiple requests issued by the same users in different time granules are called dynamic (or *historical*). A possible technique to enforce anonymity in LBS is to generalize precise location data in a request to an area including a set (called *anonymity set* [41]) of other potential issuers. An important difference between the anonymity set in service requests and the *qi-group* in databases is that while the *qi-group* includes only identities actually associated to a record in the table, the anonymity set includes also users that did not issue any request but that are potential issuers with respect to the adversary's external knowledge.

With respect to identity anonymization in generic context-aware systems, it is evident that many other kinds of context data besides location may be considered *qi*. Hence, a large amount of context data must be generalized in order to enforce anonymity. As a consequence, the granularity of generalized context data released to the service provider could be too coarse to provide the service at an acceptable quality level. In order to limit the information loss due to the generalization of context data, four different personalized anonymization models are proposed in [42]. These models allow a user to constrain the maximum level of location and profile generalization still guaranteeing the desired level of anonymity. For instance, a user could decide to constrain the maximum level of location generalization to an area of 1 km^2, while imposing no constraints on the level of generalization of her profile.

As outlined in the introduction, sensing technologies deployed in pervasive environments can be exploited by adversaries to constantly monitor the users' behavior, thus exposing the user to novel kinds of privacy attacks, like the one presented in [43]. In that work it is shown that even enforcing k-anonymity, in particular cases the attacker may recognize the actual issuer of a service request by monitoring the behavior of the potential issuers with respect to service responses. For example, consider a pervasive system of a gym, suggesting exercises on the basis of gender, age, and physiological data retrieved from body-worn sensors. Even if users are anonymous in a set of k potential issuers, the attacker can easily recognize the issuer of a particular request if she starts to use in a reasonable lapse of time a machine the system suggested to her, which was not suggested to any other potential issuer. The proposed solution relies on an intermediary entity that filters all the communications between users and service providers, calculates the privacy threats corresponding to possible alternatives suggested by the service (e.g., the next exercise to perform), and automatically filters unsafe alternatives.

A further issue to be considered is the defense against the well-known problem of *homogeneity* [44] identified in the field of databases. Homogeneity attacks can be performed if all the records belonging to a *qi*-group have the same value of sensitive information. In this case it is clear that the adversary may easily violate the users' privacy despite anonymity is formally enforced. The same problem may arise as well in context-aware services in the case an adversary recognizes that all the users in an anonymity set actually issued a request with the same value of private information. To our knowledge, a first effort to defend against such attacks in context-aware systems has been presented in [45]. That proposal aims at protecting from multiple-issuers historical attacks by applying a bounded generalization of both context data and service parameters.

6 Towards a comprehensive framework for privacy protection in context-aware systems

Based on the weaknesses emerged from the analysis of the proposed techniques, in this section we advocate the use of a combined approach to address the comprehensive issue of privacy in context awareness; we present existing proposals, and we illustrate the logical design of a framework intended to solve most of the identified problems.

On the need for a combined approach The analysis of the state-of-the-art reported in the previous sections has shown that each of the proposed approaches, even if effective in a particular scenario and under particular assumptions, fails in providing a solution to the general problem. In particular:

- cryptographic techniques for private information retrieval presented up to the time of writing are unfeasible to complex context-aware services, due to problems of bandwidth and computational resources consumption;
- protecting communication privacy between the context source and the context data consumer (e.g., the service provider) is useless in the case the context data consumer is untrusted;
- access control techniques (possibly coupled with obfuscation) are ineffective in the case the access to a service is a sensitive information by itself, since they do not protect from the disclosure of the user's identity. Moreover, they do not prevent a malicious subject to adopt reasoning techniques in order to derive new sensitive information based on data it is authorized to access;
- techniques for identity anonymity rely on the exact knowledge about the external information available to an adversary. However, especially in pervasive and mobile computing scenarios, such knowledge is very hard to obtain, and adopting worst-case assumptions about the external information leads to a significant degradation of the quality of released context data.

These observations claim for the combination of different approaches in order to protect against the different kind of attacks that can be posed to the privacy of users taking advantage of context-aware services.

Proposed techniques Proposals to combine different approaches in a common framework have been recently presented.

In [46], an architecture for privacy-conscious context aggregation and reasoning is illustrated. The proposed solution adopts client-side reasoning modules to abstract raw context data into significant descriptions of the user's situation (e.g., current activity and stereotype) that can be useful for adaptation. Release of private context information is controlled by context-aware access control policies, and the access to context information by service providers is mediated by a trusted intermediary infrastructure in charge of enforcing anonymity. Moreover, cryptographic techniques are used to protect communications inside the user trusted domain.

Papadopoulou et al. present in [47] a practical solution to enforce anonymity. In that work, no assumptions about the external knowledge available to an adversary are made; hence, the proposed technique does not formally guarantee a given anonymity level. For this reason, the anonymization technique is coupled with access control and obfuscation mechanisms in order to protect privacy in the case an adversary is able to discover the user's identity. That technique is applied using the *virtual identity* metaphor. A virtual identity is essentially the subset of context data that a user is willing to share with a third party in a given situation; in addition, since anonymity is not formally guaranteed, part

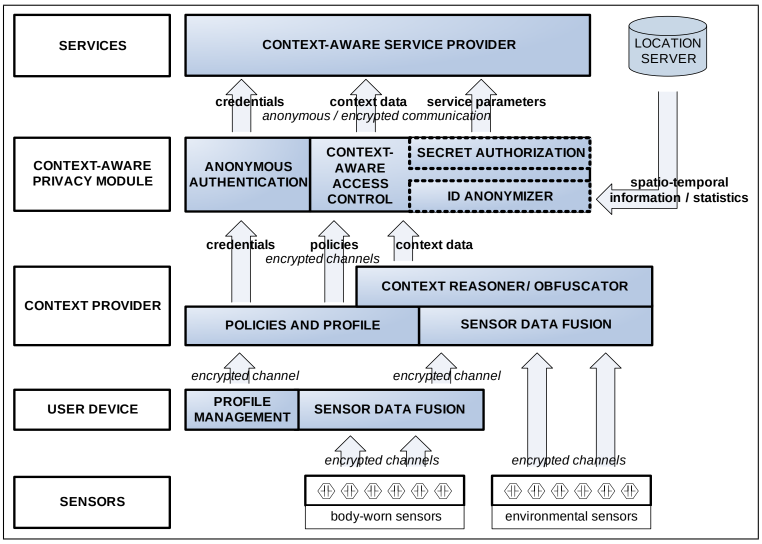


Fig. 2. The envisioned framework

of the shared context data can be obfuscated on the basis of privacy policies in order to hide some sensible details. For instance, a person could decide to share her preferences regarding shopping items and leisure activities, as well as her obfuscated location, when she is on vacation (using a *tourist* virtual identity), while hiding those information when she is traveling for work (using a *worker* virtual identity). With respect to the problem introduced by multiple requests issued by the same user, specific techniques are presented to avoid that different virtual identities can be linked to the same (anonymous) user by an adversary.

While the above mentioned works try to protect the privacy of users accessing a remote service, the *AnonySense* system [48] is aimed at supporting privacy in opportunistic sensing applications, i.e., applications that leverage opportunistic networks formed by mobile devices to acquire aggregated context data in a particular region. To reach this goal, the geographic area is logically partitioned into tiles large enough to probabilistically gain k-anonymity; i.e., regions visited with high probability by more than k persons during a given time granule. Measurements of context data are reported by mobile nodes specifying the tile they refer to and the time interval during which they were acquired. Moreover, in order to provide a second layer of privacy protection, obfuscation is applied on the sensed data by fusing the values provided by at least l nodes ($l \leq k$) before communicating the aggregated data to the application. Cryptographic techniques are used to enforce anonymous authentication by users of the system.

Towards a comprehensive framework We now illustrate how existing techniques can be extended and combined in a logical multilayer framework, which is graphically depicted in Figure 2. This framework is partially derived from the preliminary architecture described in [46]. However, the model presented here is intended to provide a more comprehensive privacy solution, addressing problems regarding sensor and profile data aggregation and reasoning (including obfuscation), context-aware access control and secret authorization, anonymous authentication, identity anonymity, and anonymous/encrypted communication. Clearly, the actual techniques to be applied for protecting privacy depend on the current context (users' situation, available services, network and environmental conditions). However, we believe that this framework is flexible enough to provide effective privacy protection in most pervasive and mobile computing scenarios. The framework is composed of the following layers:

- ***Sensors*** **layer:** This layer includes body-worn and environmental sensors that communicate context data to the upper layers through encrypted channels using energy-efficient cryptographic protocols (e.g., those based on elliptic curves [49] like in Sun SPOT sensors [50]). We assume that this layer is within the trusted domain of the user (i.e., sensors do not deliberately provide false information).
- ***User device*** **layer:** This layer is in charge of managing the user's profile information (i.e., context data that are almost static, like personal information, interests and preferences) and privacy policies. Upon update of this information by the user, the new information is communicated to the upper layer. Moreover, this layer is in charge of fusing context data provided by body-worn sensors and to communicate them in an aggregated form to the upper layer on a *per-request* basis (e.g., when those data are required by a service for performing adaptation). This layer is deployed on the user's device, which is assumed to be trusted (traditional security issues are not addressed here); communications with the upper layer are performed through encrypted channels.
- ***Context provider*** **layer:** This layer is in charge of fusing sensor data provided by the lower layers, including those provided by sensors that are not directly under the communication range of the user device. Moreover, according to the user's policies, it performs context reasoning and obfuscation for privacy and adaptation purposes, as described in [46]. It communicates user's credentials, privacy policies, and context data to the upper layer on a *per-request* basis through encrypted channels. This layer belongs to the user's trusted domain; depending on the device capabilities, it can be deployed on the user's device itself, or on another trusted machine.
- ***Context-aware privacy module*** **layer:** This layer is in charge of anonymously authenticating the user on the upper layer, and to enforce her context-aware access control policies, possibly after a phase of secret negotiation with the third party. Moreover, depending on the user's policies, it can possibly anonymize the user's identity on the basis of (either precise or statistical) trusted information received from the upper layer (e.g., spatio-temporal information about users received from a trusted location server). Protocols for anonymous/encrypted communication are adopted to provide credentials,

context data and service parameters to the upper layer. This layer belongs to the user's trusted domain. Depending on device capabilities and on characteristics of the actual algorithms it adopts (e.g., to enforce anonymity), this layer can be implemented on the user's device, on another trusted machine, or on the infrastructure of a trusted entity (e.g., the network operator).

◦ ***Services* layer:** This layer is composed of context-aware service providers and other infrastructural services (e.g., location servers). Typically, this layer is assumed not to belong to the user's trusted domain, even if particular services can be trusted by the user (e.g., a network operator location server).

7 Conclusions

Through a classification into four main categories of techniques, we have described the state of the art of privacy preservation for georeferenced context-aware services. While previous work has also proposed the combination of techniques from two or more categories, we claim that a deeper integration is needed and we propose an architecture for a comprehensive framework towards this goal. Clearly, there is still a long way to go in order to refine the architecture, work out the details of its components, implement and integrate the actual techniques, and test the framework on real applications. Moreover, there are still several other aspects, not considered in our paper, that deserve investigation. For example, since there are well-known techniques for context reasoning, they may have to be taken into account, since released context data may determine the disclosure of other context data, possibly leading to privacy leaks that were previously unidentified. Furthermore, computationally expensive techniques (e.g., those making use of ontological reasoning or complex cryptographic algorithms) pose serious scalability issues that may limit their applicability in real-world scenarios. Finally, since the access to context data of real users is generally unavailable for privacy reasons, sophisticated simulation environments are needed to evaluate the actual effectiveness of privacy preservation mechanisms in realistic situations.

Acknowledgments

This work was partially supported by National Science Foundation (NSF) under grants N. CNS-0716567 and N. IIS-0430402, and by Italian MIUR under grant InterLink II04C0EC1D.

References

1. Palen, L., Dourish, P.: Unpacking "privacy" for a networked world. In: Proceedings of the 2003 Conference on Human Factors in Computing Systems (CHI 2003), ACM (2003) 129–136
2. Lederer, S., Hong, J.I., Dey, A.K., Landay, J.A.: Personal privacy through understanding and action: five pitfalls for designers. Personal and Ubiquitous Computing **8**(6) (2004) 440–454

3. Bettini, C., Mascetti, S., Wang, X.S.: Privacy Protection through Anonymity in Location-based Services. Handbook of Database Security: Applications and Trends (2008) 509–530
4. Atallah, M.J., Frikken, K.B.: Privacy-Preserving Location-Dependent Query Processing. In: ICPS '04: Proceedings of the The IEEE/ACS International Conference on Pervasive Services, IEEE Computer Society (2004) 9–17
5. Ghinita, G., Kalnis, P., Khoshgozaran, A., Shahabi, C., Tan, K.L.: Private queries in location based services: anonymizers are not necessary. In: Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD 2008), ACM (2008) 121–132
6. Ardagna, C.A., Cremonini, M., Damiani, E., De Capitani di Vimercati, S., Samarati, P.: Location Privacy Protection Through Obfuscation-Based Techniques. In: Proceedings of the 21st Annual IFIP WG 11.3 Working Conference on Data and Applications Security (DBSec'07). Volume 4602 of Lecture Notes in Computer Science., Springer (2007) 47–60
7. Yiu, M.L., Jensen, C.S., Huang, X., Lu, H.: SpaceTwist: Managing the Trade-Offs Among Location Privacy, Query Performance, and Query Accuracy in Mobile Services. In: Proceedings of the 24th International Conference on Data Engineering (ICDE 2008), IEEE Computer Society (2008) 366–375
8. Gruteser, M., Grunwald, D.: Anonymous Usage of Location-Based Services Through Spatial and Temporal Cloaking. In: Proc. of the 1st International Conference on Mobile Systems, Applications and Services (MobiSys), USENIX Association (2003) 31–42
9. Gedik, B., Liu, L.: Protecting Location Privacy with Personalized k-Anonymity: Architecture and Algorithms. IEEE Transactions on Mobile Computing **7**(1) (2008) 1–18
10. Aggarwal, C.C.: On k-Anonymity and the Curse of Dimensionality. In: Proceedings of the 31st International Conference on Very Large Data Bases (VLDB), ACM (2005) 901–909
11. Neuman, B., Ts'o, T.: Kerberos: an authentication service for computer networks. Communications Magazine, IEEE **32**(9) (Sep 1994) 33–38
12. Freedman, M.J., Morris, R.: Tarzan: a peer-to-peer anonymizing network layer. In: CCS '02: Proceedings of the 9th ACM conference on Computer and communications security, ACM (2002) 193–206
13. Reiter, M.K., Rubin, A.D.: Anonymous web transactions with crowds. Commun. ACM **42**(2) (1999) 32–48
14. Dingledine, R., Mathewson, N., Syverson, P.: Tor: the second-generation onion router. In: SSYM'04: Proceedings of the 13th conference on USENIX Security Symposium, USENIX Association (2004) 21–21
15. Goldschlag, D., Reed, M., Syverson, P.: Onion routing. Commun. ACM **42**(2) (1999) 39–41
16. Al-Muhtadi, J., Campbell, R., Kapadia, A., Mickunas, M.D., Yi, S.: Routing Through the Mist: Privacy Preserving Communication in Ubiquitous Computing Environments. In: Proceedings of the 22 nd International Conference on Distributed Computing Systems (ICDCS'02), IEEE Computer Society (2002) 74
17. Atallah, M.J., Du, W.: Secure multi-party computational geometry. In: WADS '01: Proceedings of the 7th International Workshop on Algorithms and Data Structures, Springer-Verlag (2001) 165–179
18. Samarati, P., De Capitani di Vimercati, S.: Access Control: Policies, Models, and Mechanisms. In: Foundations of Security Analysis and Design, Tutorial Lectures. Volume 2171 of Lecture Notes in Computer Science., Springer (2001) 137–196

19. Kumar, A., Karnik, N.M., Chafle, G.: Context sensitivity in role-based access control. Operating Systems Review **36**(3) (2002) 53–66
20. Covington, M.J., Fogla, P., Zhan, Z., Ahamad, M.: A Context-Aware Security Architecture for Emerging Applications. In: Proceedings of the 18th Annual Computer Security Applications Conference (ACSAC 2002), IEEE Computer Society (2002) 249–260
21. Toninelli, A., Montanari, R., Kagal, L., Lassila, O.: Proteus: A Semantic Context-Aware Adaptive Policy Model. In: Proceedings of the 8th IEEE International Workshop on Policies for Distributed Systems and Networks(POLICY 2007), IEEE Computer Society (2007) 129–140
22. Sandhu, R., Samarati, P.: Access Control: Principles and Practice. IEEE Communications **32**(9) (1994) 40–48
23. Sandhu, R.S., Coyne, E.J., Feinstein, H.L., Youman, C.E.: Role-Based Access Control Models. IEEE Computer **29**(2) (1996) 38–47
24. Hengartner, U., Steenkiste, P.: Avoiding Privacy Violations Caused by Context-Sensitive Services. Pervasive and Mobile Computing **2**(3) (2006) 427–452
25. Brands, S.A.: Rethinking Public Key Infrastructures and Digital Certificates: Building in Privacy. MIT Press (2000)
26. Wang, C.D., Feng, L.C., Wang, Q.: Zero-Knowledge-Based User Authentication Technique in Context-aware System. Multimedia and Ubiquitous Engineering, 2007. MUE '07. International Conference on (April 2007) 874–879
27. Hengartner, U., Steenkiste, P.: Access control to people location information. ACM Trans. Inf. Syst. Secur. **8**(4) (2005) 424–456
28. Hull, R., Kumar, B., Lieuwen, D., Patel-Schneider, P., Sahuguet, A., Varadarajan, S., Vyas, A.: Enabling Context-Aware and Privacy-Conscious User Data Sharing. In: Proceedings of the 2004 IEEE International Conference on Mobile Data Management (MDM'04), IEEE Computer Society (2004) 187–198
29. Atluri, V., Shin, H.: Efficient Security Policy Enforcement in a Location Based Service Environment. In: Proceedings of Data and Applications Security XXI, 21st Annual IFIP WG 11.3 Working Conference on Data and Applications Security. Volume 4602 of Lecture Notes in Computer Science., Springer (2007) 61–76
30. Atluri, V., Chun, S.A.: A geotemporal role-based authorisation system. International Journal of Information and Computer Security **1**(1–2) (2007) 143–168
31. Corradi, A., Montanari, R., Tibaldi, D.: Context-Based Access Control Management in Ubiquitous Environments. In: Proceedings of the 3rd IEEE International Symposium on Network Computing and Applications (NCA 2004), IEEE Computer Society (2004) 253–260
32. Sacramento, V., Endler, M., Nascimento, F.N.: A Privacy Service for Context-aware Mobile Computing. In: Proceedings of the First International Conference on Security and Privacy for Emerging Areas in Communications Networks (SECURECOMM '05), IEEE Computer Society (2005) 182–193
33. Zhang, Q., Qi, Y., Zhao, J., Hou, D., Zhao, T., Liu, L.: A Study on Context-aware Privacy Protection for Personal Information. In: Proceedings of the 16th IEEE International Conference on Computer Communications and Networks (ICCCN 2007), IEEE Computer Society (2007) 1351–1358
34. Bakken, D.E., Parameswaran, R., Blough, D.M., Franz, A.A., Palmer, T.J.: Data Obfuscation: Anonymity and Desensitization of Usable Data Sets. IEEE Security & Privacy **2**(6) (2004) 34–41
35. Xiao, X., Tao, Y.: Personalized privacy preservation. In: SIGMOD '06: Proceedings of the 2006 ACM SIGMOD international conference on Management of data, ACM Press (2006) 229–240

36. Duckham, M., Kulik, L.: A Formal Model of Obfuscation and Negotiation for Location Privacy. In: Proceedings of the Third International Conference on Pervasive Computing (PERVASIVE 2005). Volume 3468 of Lecture Notes in Computer Science., Springer (2005) 152–170
37. Gandon, F.L., Sadeh, N.M.: Semantic web technologies to reconcile privacy and context awareness. J. Web Sem. **1**(3) (2004) 241–260
38. Wishart, R., Henricksen, K., Indulska, J.: Context Privacy and Obfuscation Supported by Dynamic Context Source Discovery and Processing in a Context Management System. In: Proceedings of the 4th International Conference on Ubiquitous Intelligence and Computing (UIC 2007). Volume 4611 of Lecture Notes in Computer Science., Springer (2007) 929–940
39. Sheikh, K., Wegdam, M., van Sinderen, M.: Quality-of-Context and its use for Protecting Privacy in Context Aware Systems. Journal of Software **3**(3) (2008) 83–93
40. Samarati, P.: Protecting Respondents' Identities in Microdata Release. IEEE Trans. on Knowledge and Data Engineering **13**(6) (2001) 1010–1027
41. Pfitzmann, A., Köhntopp, M.: Anonymity, unobservability, and pseudonymity - a proposal for terminology. In: Designing Privacy Enhancing Technologies: International Workshop on Design Issues in Anonymity and Unobservability. Volume 2009 of LNCS., Springer (July 2000) 1–9
42. Shin, H., Atluri, V., Vaidya, J.: A Profile Anonymization Model for Privacy in a Personalized Location Based Service Environment. Proceedings of the 9th International Conference on Mobile Data Management (MDM'08) (2008) 73–80
43. Riboni, D., Pareschi, L., Bettini, C.: Shadow Attacks to Users' Anonymity in Pervasive Computing Environments. Journal of Pervasive and Mobile Computing (To appear) DOI: 10.1016/j.pmcj.2008.04.008.
44. Machanavajjhala, A., Gehrke, J., Kifer, D., Venkitasubramaniam, M.: l-Diversity: Privacy Beyond k-Anonymity. In: Proceedings of ICDE 2006, IEEE Computer Society (2006)
45. Riboni, D., Pareschi, L., Bettini, C., Jajodia, S.: Preserving Privacy in LBS against Attacks based on Concurrent Requests. Technical Report TR 24-07, University of Milan (2007)
46. Pareschi, L., Riboni, D., Agostini, A., Bettini, C.: Composition and Generalization of Context Data for Privacy Preservation. In: Sixth Annual IEEE International Conference on Pervasive Computing and Communications (PerCom 2008), Proceedings of the Workshops, IEEE Computer Society (2008) 429–433
47. Papadopoulou, E., McBurney, S., Taylor, N., Williams, M.H., Dolinar, K., Neubauer, M.: Using User Preferences to Enhance Privacy in Pervasive Systems. In: Proceedings of the Third International Conference on Systems (ICONS 2008), IEEE Computer Society (2008) 271–276
48. Kapadia, A., Triandopoulos, N., Cornelius, C., Peebles, D., Kotz, D.: AnonySense: Opportunistic and Privacy-Preserving Context Collection. In: Proceedings of the 6th International Conference on Pervasive Computing (Pervasive 2008). Volume 5013 of Lecture Notes in Computer Science., Springer (2008) 280–297
49. Miller, V.S.: Use of Elliptic Curves in Cryptography. In: Proceedings of Advances in Cryptology (CRYPTO '85). Volume 218 of Lecture Notes in Computer Science., Springer (1986) 417–426
50. Simon, D., Cifuentes, C., Cleal, D., Daniels, J., White, D.: Java™ on the bare metal of wireless sensor devices: the squawk Java virtual machine. In: Proceedings of the 2nd International Conference on Virtual Execution Environments (VEE 2006), ACM (2006) 78–88

Pattern-Preserving *k*-Anonymization of Sequences and its Application to Mobility Data Mining

Ruggero G. Pensa[1], Anna Monreale[2], Fabio Pinelli[1], and Dino Pedreschi[2]

[1] ISTI - CNR, Area della Ricerca di Pisa,
Via Giuseppe Moruzzi, 1 - 56124 Pisa, Italy
[2] Computer Science Dep., University of Pisa,
Largo Pontecorvo, 3 - 56127 Pisa, Italy
{firstname.lastname}@isti.cnr.it

Abstract. Sequential pattern mining is a major research field in knowledge discovery and data mining. Thanks to the increasing availability of transaction data, it is now possible to provide new and improved services based on users' and customers' behavior. However, this puts the citizen's privacy at risk. Thus, it is important to develop new privacy-preserving data mining techniques that do not alter the analysis results significantly. In this paper we propose a new approach for anonymizing sequential data by hiding infrequent, and thus potentially sensible, subsequences. Our approach guarantees that the disclosed data are k-anonymous and preserve the quality of extracted patterns. An application to a real-world moving object database is presented, which shows the effectiveness of our approach also in complex contexts.

1 Introduction

In the last decade, many KDD techniques have been developed that provide new means for improving personalized services through the discovery of patterns and models which represent typical or unexpected customer's and user's behavior. The exponential growth of available personal data, as well as the refinement of data mining techniques, lead to new and intriguing possibilities. On the other hand, the collection and the disclosure of personal, often sensible, information increase the risk of citizen's privacy violation. For this reason, many recent research works have focused on privacy-preserving data mining [5, 24, 16, 18], proposing novel techniques that allow to extract knowledge while trying to protect the privacy of users and customers (or respondents) represented in the data[3]. This may involve techniques that return anonymized data mining results, or that provide anonymized datasets to the companies/research institution in charge of their analysis.

[3] In statistics, the problem has been extensively studied in the field of *statistical disclosure control*.

A major and rising field in data mining research concerns the analysis of sequence databases. User's actions as well as customer transactions are often stored together with their timestamps, making the temporal sequentiality of the events a powerful source of information. For instance, web logs provide the full activity of each website visitors during each browser session. Moreover, the spreading of mobile devices, such as mobile phone, GPS devices and RFIDs, has become a great source of spatio-temporal data. Companies and public institutions can now study the sequential behavior of their customers/citizens to improve their offers and services. A lot of advanced techniques have been investigated to extract patterns and models in databases of sequences [4, 27, 23], as well as in databases of moving objects (trajectories) [13]. For both legal and ethical reasons, the data owners (or custodians) should not compromise the privacy of their customers and users, and therefore should reveal as little as possible their personal sensible information. Hiding personal identifiers, such as personal IDs or quasi-identifiers (i.e., attributes that can be linked to external information to re-identify the individual to whom the information refers) may not be sufficient in the case of sequential data. If a small sequence of actions is easily referable to a few persons, an attacker may access to the whole action sequences involving these persons. For instance, if a malicious data user has access to the daylight city traffic data, and he knows that John Smith often goes from the commercial zone A to the general hospital B, and the sequence $A \Rightarrow B$ appears few times in the dataset, he can easily identify the entire sequence of locations crossed by John Smith during the day, and guess his daily behavior. Existing k-Anonymity techniques do not take into consideration the intrinsic sensibility of sequential data. Some other approaches have been proposed that requires that sensible sequences have to be pre-defined [2, 1]. Other approaches use collaborative data mining techniques [17], or propose to mine models instead of the data [15], but they do not ensure that sensible sequences can not be extracted.

In this paper, we propose a new technique that provides an anonymized dataset of sequences, while preserving sequential pattern mining results. We use a method which combines k-anonymity (the disclosed dataset is such that any sequence is undistinguishable with at least $k-1$ other sequences) and sequence hiding approaches. Our approach consists in a reformulation of the anonymization problem as the problem of hiding k-infrequent sequences, i.e., transforming the original sequence database in such way that the sequences with support less than k in the original dataset can not be mined any longer. In the hypothesis that an attacker knows part of the sequence belonging to a person, and that s/he also know that this person is present in the database, s/he has a probability of $1/k$ of reconstructing the entire sequence. Our approach is formally defined in the general setting of sequences of items, or events. To illustrate its effectiveness and practicality in a realistic and complex domain, we put at work our anonymization technique in the scenario of moving object data analysis, and applied it to a large-scale, real-life dataset of GPS trajectories of vehicles with on-board GPS receivers, tracked in the city of Milan, Italy. The results of our experiments, where we compare the set of sequential patterns obtained before and after the application of our anonymization technique, show that we can substantially preserve such frequent sequential patterns, while guaranteeing that the disclosed data are k-anonymous.

The rest of the paper is organized as follows. Section 2 briefly discusses the relevant related works on privacy-preserving data mining. Section 3 introduces and explains our Privacy-Preserving k-Anonymization (P2kA) framework. The algorithmic details are given in Section 4, together with explanations on a toy example consisting of a small set of sequences. The experimental results of our application to a moving object dataset are presented and discussed in Section 5. Finally, Section 6 concludes.

2 Related works

A lot of recent research works have focused on techniques for privacy-preserving data mining [5] and for privacy-preserving data publishing. Important techniques include perturbation, condensation, and data hiding with conceptual reconstruction. The first step before data publishing is to remove the personally identifying information. In [24] (and much earlier in statistics by T. Dalenius [9]), it has been shown that removing personally identifying information is not enough to protect privacy. In this work, Samarati and Sweeney propose a classification of the attributes in *quasi-identifiers* (i.e., attributes that can be linked to external information to re-identify the individual to whom the information refers, a concept that was already present in [10]), and sensitive attributes. Moreover, they propose the k-anonymity to generalize the values of quasi-identifier attributes in each record so that it is indistinguishable with at least $k-1$ other records with respect to the quasi-identifier, Recently, privacy-preserving data mining has been studied in conjunction with spatio-temporal data and trajectory mining [12, 8]. In the work presented in [3], the authors study the problem of anonymity preserving data publishing in moving objects databases. They propose the notion of $(k, \delta) - anonymity$ for moving objects databases. In particular, this is a novel concept of k-anonymity based on co-localization that exploits the inherent uncertainty of the moving objects whereabouts. The k-anonymity notion is also used in [22], where authors address privacy issues regarding the identification of individuals in static trajectory datasets. They provide privacy protection by: (1) first enforcing k-anonymity, meaning every released information refers to at least k users/trajectories, (2) then reconstructing randomly a representation of the original dataset from the anonymization. Although it has been shown that the k-anonymity framework presents some flaws and limitations [20], and that finding an optimal k-anonymization is NP-hard [6], the k-anonymity model is still practically relevant and in recent years a large research effort has been devoted to develop algorithms for k-anonymity [16, 18].

Existing work about anonymity of spatio-temporal moving points has been mainly developed in the context of location based services (LBS) [21, 26, 14, 7]. Works in [21, 26] use perturbation and obfuscation techniques to de-identify a given request or a location. In [14], anonymity is enforced on sensitive locations other than user location points or trajectories. In [7], anonymization process enforces points referring to same set of users to be anonymized together. However this work considers the anonymization of a request rather than the whole trajectory anonymization. In order to preserve the privacy for moving object data in [1] the authors propose a hiding technique. In particular, they address the problem of hiding sensitive trajectory patterns from a database of moving objects. A similar technique is used in [2], where Abul et al. address first the problem of hiding patterns that are a simple sequence of symbols and then they extend the proposed framework to the case of sequential patterns according to the classical definition [5]. A first work attacking the problem of limiting disclosure of sensitive rules by reducing their significance, while leaving unaltered or minimally affecting the significance of others, non-sensitive rules is [6]. One of the most important contributions of this paper is the proof that finding an optimal sanitization of a dataset is NP-hard. A heuristic using greedy search is thus proposed. In the work [11] the objective is to hide individual sensitive rules instead of all rules produced by some sensitive itemsets. The work in [25] proposes two distortion-based heuristic techniques for selectively hiding sensitive rules. An interesting work is presented in [15], where Jacquemont et al. propose a costless solution to privacy preserving for problems that may be stated as flow control problems, that is the case of frequent path discovery in Web sites and frequent route discovery in towns. They propose to model this flow of data in the form of a weighted automaton, for which they provide a probabilistic solution to discover frequent patterns (potentially with gaps) under constraints, without any information about the original data.

Essentially, in our work we present a new anonymization technique for preserving privacy and at same time, preserving also frequent sequential patterns (FSP) obtained by mining the anonymized data. The basic *frequent sequential pattern* problem, originally introduced in [4], is defined over a database of sequences D, where each element of each sequence is a time-stamped set of items — i.e., an *itemset*. Time-stamps determine the order of elements in the sequence. Then, the FSP problem consists in finding all the sequences that are *frequent* in the database, i.e., appear as subsequence of a large percentage of sequences of the database. Since its first definition, many algorithms for sequential patterns have been proposed, from the earliest in [4], to the more recent PrefixSpan [23] and SPADE [27].

3 Problem Definition

Let $L = \{l_1, l_2, \ldots, l_n\}$ denote a set of items (e.g, spatial locations or regions). A sequence $S = s_1 s_2 \ldots s_m$ ($s_i \in L$) is an ordered list of items, and an item can occur multiple times in a sequence. A sequence $T = t_1 t_2 \ldots t_w$ is a subsequence of S ($T \preceq S$) if there exist integers $1 \leq i_1 < \ldots < i_w \leq m$ such that $\forall 1 \leq j \leq w$ $t_j = s_{i_j}$. A sequence database $\mathcal{D}$ is a set of sequences $\mathcal{D} = \{S_1, S_2, \ldots, S_N\}$. The support of a sequence T in a database $\mathcal{D}$ is the number of sequences in the database containing T, i.e.:

$$supp_{\mathcal{D}}(T) = |\{S \ s.t. \ S \in \mathcal{D} \wedge T \preceq S\}|$$

Given a support threshold σ, a sequence T is called a σ-frequent sequential pattern in a sequence database $\mathcal{D}$ if $supp_{\mathcal{D}}(T) \geq \sigma$. The collection of all σ-frequent (sequential) patterns in $\mathcal{D}$ is denoted by $\mathcal{S}(\mathcal{D}, \sigma)$. The set of all subsequences supported by $\mathcal{D}$ is denoted by $\mathcal{S}(\mathcal{D})$.

Our goal is to provide an anomymized version of $\mathcal{D}$ that preserves as much as possible the collection of frequent patterns. We use a method which combines k-anonymity and sequence hiding approaches. Put in other words, we reformulate the anonymization problem — in the case of sequential data — as the problem of hiding k-infrequent sequences, i.e., transforming the original sequence database in such way that the sequences with support less than k in the original dataset can not be mined any longer. The disclosed dataset is such that any sequence is undistinguishable with at least $k-1$ other sequences. This goal is achieved by hiding all the subsequences which are not supported by at least k sequences in the database. Let $\mathcal{D}'$ denote the disclosed dataset. Given a positive integer k, the disclosed dataset $\mathcal{D}'$ is such that

$$\sum_{T \in \mathcal{S}(\mathcal{D}')} \delta[supp_{\mathcal{D}}(T) < k] \cdot supp_{\mathcal{D}'}(T) = 0$$

where $\delta[condition]$ is the Dirichlet function (which is equal to 1 if *condition* is true, 0 otherwise). In this paper we consider that any infrequent subsequence of items can potentially lead to the identification of the user (respondent). Thus, we do not need to specify any sensible subsequence preliminarily, as in [2, 1]. Moreover, we want to preserve frequent pattern mining results, in order to let the analysts investigate over frequent and interesting/unexpected behavior. The optimal **pattern-preserving k-anonymization problem** can be formulated as follows:

Definition 1 (optimal P2kA problem). *Given a sequence database $\mathcal{D}$, and a positive integer k, find a database $\mathcal{D}'$ such that*

1. *$\mathcal{D}'$ is k-anonymous, i.e.:*

$$\sum_{T \in \mathcal{S}(\mathcal{D}')} \delta[supp_{\mathcal{D}}(T) < k] \cdot supp_{\mathcal{D}'}(T) = 0$$

2. *the collection of all k-frequent pattern in $\mathcal{D}$ is preserved, i.e.:*

$$\begin{array}{l} \mathcal{S}(\mathcal{D}', k) = \mathcal{S}(\mathcal{D}, k) \\ \forall T \in \mathcal{S}(\mathcal{D}', k) \ supp_{\mathcal{D}'}(T) = supp_{\mathcal{D}'}(T) \end{array}$$

In this paper we present an algorithm which assures that (i) $\mathcal{D}'$ is k-anonymous and (ii) $\mathcal{S}(\mathcal{D}', k)$ and $\mathcal{S}(\mathcal{D}, k)$ are "similar". In particular the second condition of Definition 1 becomes:

$$\begin{array}{l} \mathcal{S}(\mathcal{D}', k) \subseteq \mathcal{S}(\mathcal{D}, k) \\ \forall T \in \mathcal{S}(\mathcal{D}', k) \ supp_{\mathcal{D}'}(T) \simeq supp_{\mathcal{D}'}(T) \end{array}$$

Algorithm 1: BF-P2kA($\mathcal{D}$, k)

```
Input: A sequence database $\mathcal{D}$, a minimum support threshold $k$
Output: A $k$-anonymous sequence database $\mathcal{D}'$
$\mathcal{PT} = PrefixTreeConstruction(\mathcal{D})$;
$\mathcal{PT}' = PTAnonymization(\mathcal{PT}, k)$
$\mathcal{D}' = SequenceGeneration(\mathcal{PT}')$;
return $\mathcal{D}'$
```

Algorithm 2: PTAnonymization($\mathcal{PT}$, k)

```
Input: A prefix tree $\mathcal{PT}$, a minimum support threshold $k$
Output: A $k$-anonymous prefix tree $\mathcal{PT}'$
$\mathcal{L}_{cut} = \emptyset$;
foreach $n$ in $Root(\mathcal{PT}).children$ do
    $\mathcal{L}_{cut} = \mathcal{L}_{cut} \cup TreePruning(n, \mathcal{PT}, k)$;
end
$\mathcal{PT}' = TreeReconstruction(\mathcal{PT}, \mathcal{L}_{cut})$;
return $\mathcal{PT}'$
```

In the experimental section (see Section 5) we will express this similarity in terms of two measures which quantify how much the pattern support changes, and how many frequent pattern we miss. As a preliminary step towards an "optimal" algorithm, we will show that our algorithm provides good results in term of pattern similarity (see Section 5), and guarantees that the disclosed dataset is k-anonymous.

4 The BF-P2kA algorithm

In this section we present our *BF-P2kA* (Brute Force Pattern-Preserving k-Anonymization) algorithm (Algorithm 1), which allows to anonymize a dataset of sequences $\mathcal{D}$. Our approach consists of three steps. During the first step, the sequences in the input dataset $\mathcal{D}$ are used to build a prefix tree $\mathcal{PT}$. The second step, given a minimum support threshold k, anonymizes the prefix tree. This means that sequences whose support is less than k are pruned from the prefix tree. Then part of these infrequent sequences is re-appended in the prefix tree. The third and last step post-process the anonymized prefix tree, as obtained in the previous step, to generate the anonymized dataset of sequences $\mathcal{D}'$.

Step I: Prefix Tree Construction The first step of the *BF-P2kA* algorithm (Algorithm 1) is the construction of a prefix tree $\mathcal{PT}$, given a list of sequences $\mathcal{D}$. The created prefix tree is a more compact structure than a list of sequences. It is defined as a triplet $\mathcal{PT} = (\mathcal{N}, \mathcal{E}, Root(\mathcal{PT}))$, where $\mathcal{N}$ is a finite set of labeled nodes, $\mathcal{E}$ is a set of edges and $Root(\mathcal{PT}) \in \mathcal{N}$ is a fictitious node and represents the root of the tree. Each node of the tree (except the root) has exactly one parent and it can be reached through a path, which is a sequence of

Algorithm 3: PrefixTreeConstruction($\mathcal{D}$)

```
Input: A sequence database $\mathcal{D}$
Output: A prefix tree $\mathcal{PT}$
foreach $T$ in $\mathcal{D}$ do
    $LP = LongestPrefixSearch(Root(\mathcal{PT}), T)$;
    Append $T$ to $LP$;
    foreach $v$ in $LP$ do
        $v.support = v.support + supp_{\mathcal{D}}(T)$;
    end
    foreach $v$ in $T \setminus LP$ do
        $v.support = supp_{\mathcal{D}}(T)$;
    end
end
return $\mathcal{PT}$
```

edges starting with the root node. An example of path for the node d (denoted $\mathcal{P}(d, \mathcal{PT})$) is the following:

$$\mathcal{P}(d, \mathcal{PT}) = (Root(\mathcal{PT}), a), (a, b), (b, c), (c, d).$$

Each node $v \in \mathcal{N}$, except $Root(\mathcal{PT})$, has entries in the form $\langle id, item, support, children\rangle$ where:

– *id* is the identifier of the node v
– *item* represents an item of a sequence
– *support* is the support of the sequence represented by the path from $Root(\mathcal{PT})$ to v
– *children* is the list of child nodes of v.

The *PrefixTreeConstruction* algorithm (see Algorithm 3) for each sequence of items T searches in $\mathcal{PT}$ the path which corresponds to the longest prefix of the sequence T. Next, it appends, to the last node of the longest prefix found, a branch which represents the remaining elements of T, updating the involved node attributes accordingly. In particular, it updates the support of each node belonging to the common prefix by adding the support of the sequence T in $\mathcal{D}$, and sets the support of the remaining nodes to $supp_{\mathcal{D}}(T)$.

Step II: Prefix Tree Anonymization The main phase of our approach is the second one. This phase is described by the *Tree Anonymization* Algorithm (Algorithm 2). Before describing this algorithm we introduce some notions which are needed to better explain our method.

Definition 2 (minimum prefix). *Let $S = s_1s_2 \dots s_n$ and $T = t_1t_2 \dots t_k$ be two sequences such that T is a subsequences of S and s_p is the first item of S such that $T \preceq s_1s_2 \dots s_p$. The sequence $S' = s_1 \dots s_p$ is the* minimum prefix *of S containing the sub-sequence T.*

Algorithm 4: TreePruning(n, $\mathcal{PT}$, k)

```
Input: A node $n$, a prefix tree $\mathcal{PT}$, a minimum support threshold $k$
Output: A list of infrequent sequences $\mathcal{L}_{cut}$
$\mathcal{L}_{cut} = \emptyset$;
if $n.support < k$ then
    $\mathcal{L}_{cut}$ = the set of all sequences in $PathTree(\mathcal{PT}, n)$;
    foreach $j \in \mathcal{P}(n, \mathcal{PT})$ do
        $j.support = j.support - n.support$;
    end
    $\mathcal{PT} = \mathcal{PT} \setminus$ the subtree induced by $n$;
else
    foreach $j \in n.children$ do
        $\mathcal{L}_{cut} \cup TreePruning(j, \mathcal{PT}, k)$;
    end
end
return $\mathcal{L}_{cut}$
```

Example 1. Let us consider the sequences

$$S = ABCDECDF$$
$$T = ACD$$

The sequence $S' = ABCD$ is the minimum prefix of S containing the subsequence T.

Definition 3 (path tree). *Let $\mathcal{PT}$ be a prefix tree, let n be a node in the prefix tree $\mathcal{PT}$. The* path tree *of n in $\mathcal{PT}$ (denoted by $PathTree(\mathcal{PT}, n)$) is the subtree induced by the set of nodes belonging to $\mathcal{P}(n, \mathcal{PT})$ plus the subtree induced by n.*

We recall now the well-known notions of Levenshtein distance [19] and Longest Common Subsequence, which are used in our algorithm.

Definition 4 (Levenshtein distance). *Let S and T be two sequences. The* ***Levenshtein (edit) distance*** *between S and T is given by the minimum number of operations needed to transform a sequences into the other, where an operation is an insertion, deletion, or substitution of a single element.*

Definition 5. *Let $\mathcal{T}$ be a set of sequences. The* ***Longest Common Subsequence*** *(LCS) is the longest subsequence common to all sequences in $\mathcal{T}$.*

The first step of the Algorithm 2 is the pruning of the prefix tree with respect to the minimum support threshold given in input. This operation is executed thanks to the *TreePruning* function (see Algorithm 4). Indeed, this function modifies the tree by pruning all the infrequent subtrees and updating the support of the path to the last frequent node. In particular, it visits the tree and, when the support of a given node n is less than the minimum support threshold k, it computes all the sequences represented by the paths which contain

Algorithm 5: TreeReconstruction($\mathcal{PT}$, $\mathcal{L}_{cut}$)

```
Input: A prefix tree $\mathcal{PT}$, a list of infrequent sequences $\mathcal{L}_{cut}$
Output: An anonymized reconstructed prefix tree $\mathcal{PT}'$
foreach distinct $S \in \mathcal{L}_{cut}$ do
    $cand = ClosestLCS(S, \mathcal{PT})$;
    $L$ = the set of nodes in $\mathcal{PT}$ belonging to the first minimum prefix
    containing cand;
    if $L$ is not empty then
        foreach $node \in L$ do
            $node.support = node.support + supp_{\mathcal{L}_{cut}}(S)$;
        end
    end
end
return $\mathcal{PT}$
```

the node n and which start from the root and reach the leaves of the sub-tree with root n. Note that for construction each node of this sub-tree has support less than k. All the computed sequences and their supports are inserted in to the list $\mathcal{L}_{cut}$. Next, the subtree with root n is cut from the tree. Therefore, the procedure *TreePruning* returns a pruned prefix tree and the list $\mathcal{L}_{cut}$. After the pruning step, the algorithm redistributes the infrequent sequences in $\mathcal{L}_{cut}$ into the pruned tree, using the *TreeReconstruction* function (see Algorithm 5). In particular, for each infrequent sequence S in $\mathcal{L}_{cut}$, it computes the LCS between S and every sequence represented by the tree. Suppose that T is the sequence such that the computed LCS is subsequence of T. Thus, the *TreeReconstruction* function selects the path of the tree that represents the minimum prefix of T containing the LCS, and increases the support of the related nodes by adding the support of S in $\mathcal{L}_{cut}$. If there are more LCSs having the same length, the function *ClosestLCS* function returns the LCS and the sequence in $\mathcal{PT}$ such that the Levenshtein distance between them is minimum. This choice allows to increase the support of a limited set of nodes not belonging to the LCS, thus reducing the noise.

Step III: Generation of anonymized sequences *PTAnonymization* algorithm returns an anonymized prefix tree, i.e., a prefix tree where only k-frequent subsequences are represented. The third step our method allows to generate the anonymized dataset $\mathcal{D}'$. This phase is performed by the *SequenceGeneration* procedure, which visits the anonymized prefix tree and generates all the represented sequences. Of course, while a sequence is generated the *Sequences-Generation* procedure considers the support of this sequence.

We show now that (i) our approach guarantees that the disclosed dataset $\mathcal{D}'$ is k-anonymous (i.e., patterns whose support is less than k in the original dataset $\mathcal{D}$ are not represented in $\mathcal{D}'$) and (ii) the set of sequential patterns in $\mathcal{D}'$ is a subset of those in $\mathcal{D}$.

s_1	A B C D E F
s_2	A B C D E F
s_3	A B C D E F
s_4	A D E F
s_5	A D E F
s_6	A D E F
s_7	B K S
s_8	B K
s_9	B K
s_{10}	D E J F

(a) A dataset of sequences

s'_1	A B C D E F
s'_2	A B C D E F
s'_3	A B C D E F
s'_4	A D E F
s'_5	A D E F
s'_6	A D E F
s'_7	A D E F
s'_8	B K
s'_9	B K
s'_{10}	B K

(b) Anonymized dataset of sequences

Fig. 1. A toy example

Theorem 1. *Let $\mathcal{D}$ be a dataset of sequences. Given a minimum support threshold k, the dataset $\mathcal{D}'$ returned by Algorithm 1 satisfies the following conditions:*

1. *$\mathcal{D}'$ is k-anonymous, i.e.:*

$$\sum_{T \in \mathcal{S}(\mathcal{D}')} \delta[supp_{\mathcal{D}}(T) < k] \cdot supp_{\mathcal{D}'}(T) = 0$$

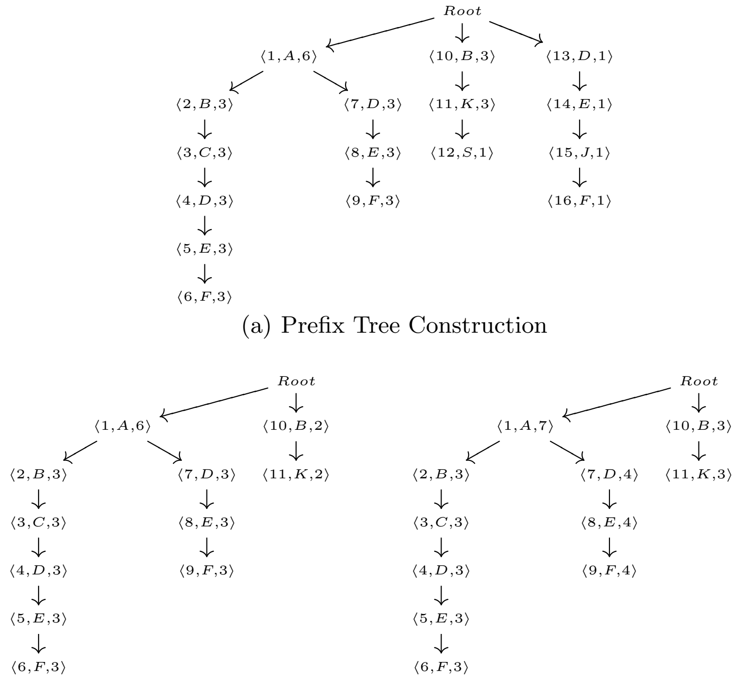
2. *$\mathcal{S}(\mathcal{D}', k) \subseteq \mathcal{S}(\mathcal{D}, k)$*

where $\mathcal{S}(\mathcal{D}, k)$ and $\mathcal{S}(\mathcal{D}', k)$ are the collections of k-frequent patterns respectively in $\mathcal{D}$ and $\mathcal{D}'$

Proof. (sketch)

1. By construction, the pruning step in Algorithm 2 prunes all the subtrees with support less than k, then the prefix tree $\mathcal{PT}$ only contains k-frequent sequences. Nevertheless, the reconstruction step (see Algorithm 5) does not change the tree structure of $\mathcal{PT}$, it only increases the support of existing sequences which are already k-frequent in $\mathcal{D}$. In conclusion, at the end of the second step of Algorithm 1, the sequential patterns which are represented in $\mathcal{PT}'$ are at least k-frequent in $\mathcal{D}$.
2. At the end of the pruning step in Algorithm 2, all infrequent branches in $\mathcal{PT}$ are cut off. However, this could also imply that some k-frequent sequential patterns are pruned out, if they are only supported by multiple infrequent paths in the prefix tree $\mathcal{PT}$. Then, the prefix tree $\mathcal{PT}$ contains a subset of the $\mathcal{S}(\mathcal{D}, k)$. Moreover, as already stated, during the reconstruction step the tree structure of $\mathcal{PT}$ is unchanged, i.e., patterns represented in $\mathcal{PT}'$ were still represented in $\mathcal{PT}$ after the pruning step. Finally, the set of sequential patterns supported by $\mathcal{D}'$ is a subset of those supported by $\mathcal{D}$.

Even if our approach does not assure that $\mathcal{S}(\mathcal{D}', k) = \mathcal{S}(\mathcal{D}, k)$, we will show in Section 5 that the difference between the two sets can be very small in practice.



(a) Prefix Tree Construction

(b) Pruned Prefix Tree

(c) Anonymized Prefix Tree

Fig. 2. Prefix tree processing

4.1 A running example

We present now an example which shows how our approach works. We consider the dataset of sequences in Figure 1(a) and a minimum support threshold equal to 2. During the first phase of our method the *PrefixTreeConstruction* algorithm builds the prefix tree depicted in Figure 2(a), which represents the sequences in a more compact way.

During the anonymization step, the prefix tree is modified by the *TreePruning* procedure with respect to the minimum support threshold. In particular, this procedure searches the tree for all nodes with support less than 2:

```
<12, S, 1>  <13, D, 1>.
```

Next, it selects the paths that contain these nodes and which start from the root and reach each leaves belonging to the subtrees of these nodes. Then, it generates all the sequences represented by these paths and inserts them into the list $\mathcal{L}_{cut}$:

```
(B K S, 1)    (D E J F, 1).
```

Finally, the *TreePruning* procedure eliminates from the tree the subtrees induced by the infrequent nodes listed above and updates the support of the remaining nodes. The prefix tree obtained after the pruning step is shown in the Figure 2(b).

The infrequent sequences within $\mathcal{L}_{cut}$ are then redistributed in this way:

1. `(B K S, 1)` increases the support of the following nodes
   `<10, B, 2>   <11, K, 2>`
   and thus we obtain
   `<10, B, 3>   <11, K, 3>`
2. `(D E J F, 1)` increases the support of the following nodes of the tree
   `<1, A, 6>    <7, D, 3>    <8, E, 3>    <9, F, 3>`
   therefore we obtain
   `<1, A, 7>    <7, D, 4>    <8, E, 4>    <9, F, 4>`.

The prefix tree obtained after the anonymization step is shown in Figure 2(c). Finally, the *SequencesGeneration* procedure provides the anonymized sequence dataset shown in Figure 1(b).

5 Experiments & Results

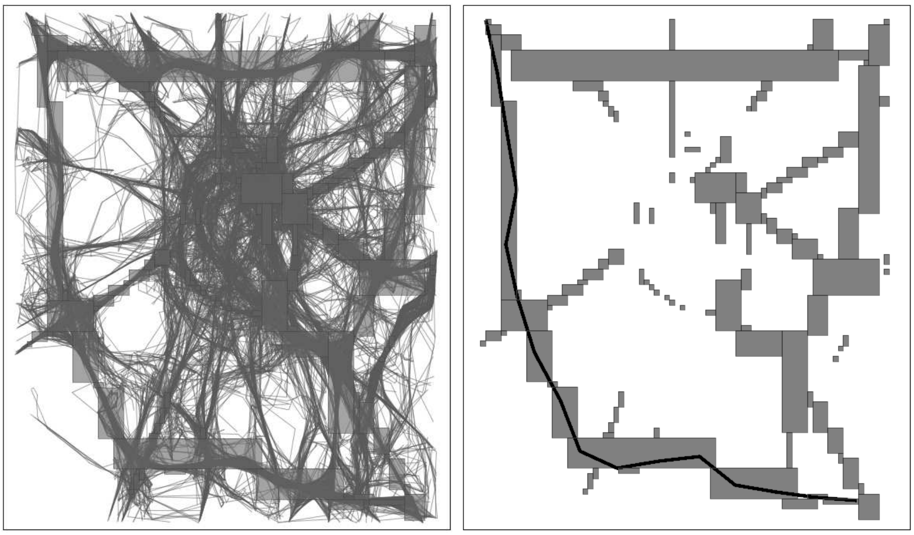
In this section, we present an application to a moving objects dataset. Object trajectories are first transformed into sequences of crossed locations, and then processed with our anonymization approach. In the following, we discuss the results over multiple instances of the original data, for different anonymization degrees.

5.1 Data Preparation

In this section, we explain the procedure used to obtain the input datasets. We got a set of GPS trajectories of cars from the european project *GeoPKDD*[4] that cover a week of traffic in Milan. Essentially, each trajectory is a sequence of pairs of coordinates x and y with relative timestamp. Obviously, performing our algorithm over sequences of points is practically useless because it is impossible to find a set of points that exactly matches enough times for being considered frequent with respect to any values of k. Thus, to overcome this problem, we use the definition of Regions of Interest given in [13], where the authors discretize the working space through a regular grid with cells of small size. Then the density of each cell is computed by considering each single trajectory and incrementing the density of all the cells that contain any of its points. Finally a set of RoI's is extracted by means of a simple heuristics using a density threshold.

As a result, a set of Roi's provides a coverage of dense cells through different sized, disjoint, rectangular regions with some form of local maximality. In particular, for each region they consider the average density of its cells, instead of its overall density (which is generally higher), and larger rectangles are preferred only if they add dense regions.

[4] http://www.geopkdd.eu



(a) (b)

Fig. 3. Trajectories and regions.

Once the set of RoI's has been extracted, we preprocess all the input trajectories translating each one from a sequence of points to a sequence of RoI's. The order of visit is maintained by means of timestamps. An example of this simple procedure of translation is shown in Fig. 3 — on the left we can see all the trajectories and a set of RoI's extracted; on the right we show a trajectory and we evidence which RoI's it crosses. This new dataset represents the input dataset for the anonymization algorithm.

The datasets used in our experiments are built using all the trajectories in the dataset described above with different density thresholds. These values have been chosen in order to obtain an adequate number of RoI's, since low density values correspond to few big regions, and higher values produce few small regions. In that way, we obtain different sets of RoI's meaning different sets of items in the input sequences. Table 1 summarizes the datasets used in our experiments. Notice that the number of trajectories is different among the datasets because we lose those trajectories that do not cross any region.

5.2 Results and discussion

Since our goal is to preserve local patterns (i.e., local subsequences) as much as possible, we compare the collections of pattern extracted before and after the anonymization process. To measure the similarity between two collection of patterns, we define two metrics:

Dataset	Density threshold	N. of Regions	N.of Trajectories	Avg. Length
1	0.01	113	82341	8.327
2	0.035	31	28663	9.152
3	0.037	21	24995	7.519
4	0.038	16	23744	6.239
5	0.039	10	10604	6.687
6	0.04	8	9213	6.863

Table 1. Input parameter

- SIM1 (Frequent Pattern Support Similarity): defined as

$$\frac{1}{|\mathcal{S}(\mathcal{D}',\sigma)|}\sum_{s\in\mathcal{S}(\mathcal{D}',\sigma)}\frac{\min\{\mathit{freq}(s,\mathcal{D}'),\mathit{freq}(s,\mathcal{D})\}}{\max\{\mathit{freq}(s,\mathcal{D}'),\mathit{freq}(s,\mathcal{D})\}}$$

- SIM2 (Frequent Pattern Collection Size Similarity): defined as

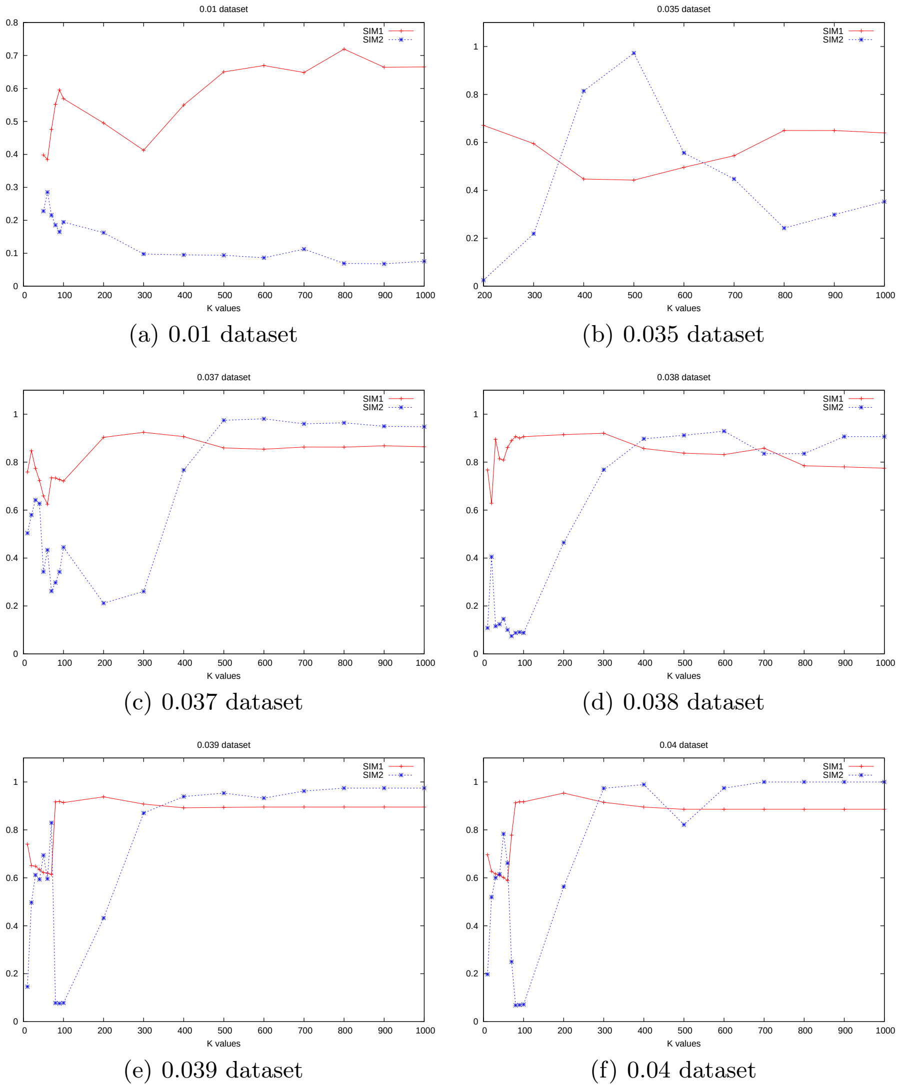
$$\frac{\min\{|\mathcal{S}(\mathcal{D}',\sigma)|,|\mathcal{S}(\mathcal{D},\sigma)|\}}{\max\{|\mathcal{S}(\mathcal{D}',\sigma)|,|\mathcal{S}(\mathcal{D},\sigma)|\}}$$

All these measures are defined between 0 and 1. When two collections of subsequences are identical, the two measures are all equal to 1.

Our experiments were conducted as follows: we first anonymized the six datasets using values of k between 10 and 1000. Then, for each value of k we compared the collection of frequent patterns extracted from the original dataset and the collection extracted from the k-anonymized dataset. In all these experiments, we used PrefixSpan [23] and the minimum support threshold was set to k.

In Figure 4 we report the results of all the experiments. We were unable to compare results for $k < 50$ and $k < 200$ for the two first datasets, since the related pattern collections are too huge and then untractable. As expected, some frequent patterns in $\mathcal{D}$ are missing in $\mathcal{D}'$. This is more evident in the first two datasets (see Figure 4(a) and 4(b)), while for higher density thresholds (Figure 4(c) to 4(f)) the value of SIM2 is closer to the maximum. This is in part due to the fact that when data are sparser, the anonymization algorithm tends to prune more sequences. Concerning the effective support similarity (SIM1), the results show that the higher the k threshold, the more similar the relative frequencies. Moreover, the SIM1 measure is quite high in general (the only exception is for the 0.035 dataset).

It is interesting to notice that, for some datasets, it is possible to identify an "optimum" minimum value of k. For instance, if we look at the similarity measures for the last dataset (see Figure 4(f)), $k = 300$ that preserves the number of frequent patterns, as well as their support. For the first dataset (see Figure 4(a)), two good choices are $k = 100$ and $k = 500$. This may possibly help the data publisher in deciding of a suitable value of k. A possible methodology would consist in finding the best tradeoff (w.r.t. the application) between the anonymization degree and the number of preserved patterns.



(a) 0.01 dataset

(b) 0.035 dataset

(c) 0.037 dataset

(d) 0.038 dataset

(e) 0.039 dataset

(f) 0.04 dataset

Fig. 4. Values of SIM1 and SIM2 for different location datasets

6 Conclusion and future work

In this paper, we introduced a new approach for anonymizing sequential datasets. Our approach provides k-anonymous data generalizing the sequence hiding approach. Through an experiment of application to a real-life mobility dataset, we showed that the proposed technique preserves sequential pattern mining results both in terms of number of extracted patterns and their support.

Further research will investigate over new approaches to preserve pattern mining results also in other hard contexts, such as sparse datasets or long sequences. One possible strategy might require the usage of a different and more compact data structure, instead of the prefix tree which is used here. Moreover, another investigation possibility could be oriented to a relaxed privacy constraint. Instead of guaranteeing the full satisfaction of k-anonymity, we could enable better pattern mining results despite of a less aggressive (and slightly more risky) pruning step. Concerning the application to mobility data, our approach does not consider yet the precious information carried by temporal annotations as well as the geographical proximity of locations/regions. A deep research effort will be undertaken to investigate on the possible extension of our approach towards a comprehensive privacy-preserving spatio-temporal framework.

Acknowledgements The authors wish to thank the anonymous reviewers for their precious comments, and Roberto Trasarti for his technical support. This work has been carried out within the European project GeoPKDD, which acknowledges the financial support of the Future and Emerging Technologies (FET) programme within the Sixth Framework Programme for Research of the European Commission, under FET-Open contract number 014915.

References

1. Osman Abul, Maurizio Atzori, Francesco Bonchi, and Fosca Giannotti. Hiding sensitive trajectory patterns. In *ICDM Workshops*, pages 693–698. IEEE Computer Society, 2007.
2. Osman Abul, Maurizio Atzori, Francesco Bonchi, and Fosca Giannotti. Hiding sequences. In *ICDE Workshops*, pages 147–156. IEEE Computer Society, 2007.
3. Osman Abul, Francesco Bonchi, and Mirco Nanni. Never walk alone: Uncertainty for anonymity in moving objects databases. In *ICDE*, pages 376–385. IEEE, 2008.
4. R. Agrawal and R. Srikant. Mining sequential patterns. In *Proceedings of ICDE*, 1995.
5. R. Agrawal and R. Srikant. Privacy-preserving data mining. In *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data (SIGMOD 2000)*, pages 439–450, 2000.
6. M. Atallah, E. Bertino, A. Elmagarmid, M. Ibrahim, and V. S. Verykios. Disclosure limitation of sensitive rules. In *Proceedings of the 1999 IEEE Knowledge and Data Engineering Exchange Workshop (KDEX'99)*, pages 45–52, 1999.
7. Claudio Bettini, Xiaoyang Sean Wang, and Sushil Jajodia. Protecting privacy against location-based personal identification. In *Proceedings VLDB Workshop SDM 2005*, volume 3674 of *LNCS*, pages 185–199. Springer, 2005.

8. Francesco Bonchi, Yücel Saygin, Vassilios S. Verykios, Maurizio Atzori, Aris Gkoulalas-Divanis, Selim. V. Kaya, and Erkay Savas. *Privacy in Spatiotemporal Data Mining*, pages 297–329. Springer Verlang, 2008.
9. Tore Dalenius. The invasion of privacy problem and statistics production — an overview. *Statistik Tidskrift*, 12:213–225, 1974.
10. Tore Dalenius. inding a needle in a haystack — or identifying anonymous census record. *Journal of Official Statistics*, 2(3):329–336, 1986.
11. Elena Dasseni, Vassilios S. Verykios, Ahmed K. Elmagarmid, and Elisa Bertino. Hiding association rules by using confidence and support. In Ira S. Moskowitz, editor, *Information Hiding*, volume 2137 of *LNCS*, pages 369–383. Springer, 2001.
12. Dino Pedreschi Fosca Giannotti, editor. *Mobility Data Mining and Privacy: Geographic Knowledge Discovery.* Springer Verlang, 2008.
13. Fosca Giannotti, Mirco Nanni, Fabio Pinelli, and Dino Pedreschi. Trajectory pattern mining. In Pavel Berkhin, Rich Caruana, and Xindong Wu, editors, *KDD*, pages 330–339. ACM, 2007.
14. M. Gruteser and X. Liu. Protecting privacy in continuous location-tracking applications. *IEEE Security & Privacy Magazine*, 2(2):28–34, 2004.
15. Stéphanie Jacquemont, François Jacquenet, and Marc Sebban. Sequence mining without sequences: A new way for privacy preserving. In *ICTAI*, pages 347–354. IEEE Computer Society, 2006.
16. Roberto J. Bayardo Jr. and Rakesh Agrawal. Data privacy through optimal k-anonymization. In *ICDE*, pages 217–228. IEEE Computer Society, 2005.
17. Seung-Woo Kim, Sanghyun Park, Jung-Im Won, and Sang-Wook Kim. Privacy preserving data mining of sequential patterns for network traffic data. *Inf. Sci.*, 178(3):694–713, 2008.
18. Kristen LeFevre, David J. DeWitt, and Raghu Ramakrishnan. Mondrian multidimensional k-anonymity. In Ling Liu, Andreas Reuter, Kyu-Young Whang, and Jianjun Zhang, editors, *ICDE*, page 25. IEEE Computer Society, 2006.
19. Vladimir I. Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady*, 10(8):707–710, 1966.
20. Ashwin Machanavajjhala, Daniel Kifer, Johannes Gehrke, and Muthuramakrishnan Venkitasubramaniam. *L*-diversity: Privacy beyond *k*-anonymity. *TKDD*, 1(1), 2007.
21. S. Menon, S. Sarkar, and S. Mukherjee. Maximizing accuracy of shared databases when concealing sensitive patterns. *Information Systems Research*, 16(3):256–270, 2005.
22. Mehmet Ercan Nergiz, Maurizio Atzori, and Yucel Saygin. Perturbation-driven anonymization of trajectories. Technical Report 2007-TR-017, ISTI-CNR, Pisa, Italy, 2007. 10 pages.
23. J. Pei et al. Prefixspan: Mining sequential patterns by prefix-projected growth. In *ICDE*, pages 215–225, 2001.
24. P. Samarati and L. Sweeney. Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression. Technical report, SRI International, March 1998.
25. Yücel Saygin, Vassilios S. Verykios, and Chris Clifton. Using unknowns to prevent discovery of association rules. *SIGMOD Record*, 30(4):45–54, 2001.
26. V. S. Verykios, E. Bertino, I. N. Fovino, L. P. Provenza, Y. Saygin, and Y. Theodoridis. State-of-the-art in privacy preserving data mining. *ACM SIGMOD Record*, 33(1):50–57, 2004.
27. M. J. Zaki. Spade: An efficient algorithm for mining frequent sequences. *Machine Learning*, 42(1/2):31–60, 2001.

On the Impact of User Movement Simulations in the Evaluation of LBS Privacy-Preserving Techniques

Sergio Mascetti[1], Dario Freni[1], Claudio Bettini[1],
X. Sean Wang[2], and Sushil Jajodia[3]

[1] DICo, Università di Milano
[2] Dept. of CS, University of Vermont
[3] CSIS, George Mason University

Abstract. The evaluation of privacy-preserving techniques for LBS is often based on simulations of mostly random user movements that only partially capture real deployment scenarios. We claim that benchmarks tailored to specific scenarios are needed, and we report preliminary results on how they may be generated through an agent-based context-aware simulator. We consider privacy preserving algorithms based on spatial cloaking and compare the experimental results obtained on two benchmarks: the first based on mostly random movements, and the second obtained from the context-aware simulator. The specific deployment scenario is the provisioning of a friend-finder-like service on weekend nights in a big city. Our results show that, compared to the context-aware simulator, the random user movement simulator leads to significantly different results for a spatial-cloaking algorithm, under-protecting in some cases, and over-protecting in others.

1 Introduction

Location-based services (LBS) are often cited as killer applications for the latest GPS-equipped 3G phones. These phones are slated to be massively distributed in 70 countries. While car navigation and identification of nearest points of interest are already widely used services, more interest are generating the so-called *friend-finder* services as a class of LBS that will change once more our way to interact. A friend-finder service reveals to a participating user the presence of other close-by participants belonging to a particular group (friends is only one example), possibly showing their position on a map. From a technical point of view, in contrast to services that find nearest points of interests, this service is characterized by a sequence of LBS requests instead of single ones, since a user may want to periodically check, while moving or even while staying in the same place, for close-by participants.

Sociological studies have shown that a large number of users perceive the release of their precise location, as part of a LBS request, as a possible privacy threat [1]. Considering friend-finder services it is easy to identify two types of privacy threats: a) the association of the identity of the user with the specific group of persons he is interested in may reveal her religious, sexual, or political orientation, and b) the association of the identity of the user with her precise location may reveal what kind of places she has been to, or that she has not been where she was supposed to be at that time.

As formally shown in [2] the likelihood of a privacy violation, and consequently the defense techniques to be applied, strongly depend on the knowledge that an adversary may have. In the friend-finder service the service provider (SP) may not be trusted or the communication channels may be insecure; then, the adversary's knowledge may include the precise identity and location information submitted with each request, and both the above privacy threats would become real privacy breaches. The substitution of identities with pseudonyms does not entirely solve the problem, if, for example, the adversary happens to know who is at the location at the time reported in the request (e.g., in the case the issuer of the request uses a fidelity card at a store). In some cases, the adversary may also be able to recognize sequences of requests as issued by the same anonymous user (e.g., by observing the same pseudonym or by spatio-temporal tracking) and use this information to re-identify the issuer.

Several defense techniques against both threats under different adversary models have been proposed, and may be applied to the friend-finder service; however, current proposals very rarely have formal assessments of the provided privacy preservation, and are generally supported by experimental results based either on real datasets of questionable significance for real LBS services (i.e., trucks or school bus traces) or on data simulations based on mostly random user movements that hardly match the specific deployment scenario of a LBS service.

In order to understand if the use of simulations based on mostly random user movements may be a real problem, or if it is actually useful and safe to use these simulations, we considered a typical deployment scenario for a friend-finder service: a large number of young people using the service on a weekend night in a large city like Milan, Italy. We performed a deep study, using different sources, including on-line surveys, of the parameters characterizing this scenario. We then used the Brinkhoff simulator [3], widely used in testing LBS privacy preservation, to generate, based on the parameters, a first dataset of user movements. A second dataset was created with a personalized version of the Siafu agent-based context-aware simulator [4] which is able to capture much more details of our scenario. Then, based on a common metric for privacy and quality of service evaluation, we run a large number of tests on both datasets, considering different abilities of re-identification by the adversary, as well as different privacy preserving techniques.

Our results consistently show that (i) in some cases the evaluation on random movement simulations leads to the definition of overprotective techniques and (ii) in other cases, the techniques that are shown to meet privacy requirements based on those simulations do not meet them when tested with more realistic context-aware simulations.

We focus our technical treatment on protecting the association of the user with the request he has issued (e.g., with the group of people he is interested in, as in threat (a) described above), even if we believe that our arguments can be easily extended to techniques only aimed to protect the location.

Related work

We are not aware of related work in this area considering specifically the relevance of realistic simulations in LBS. There are however several studies on user movements with impact on many different application areas including epidemiology, transportation, computer networks, marketing, as well as LBS. A very interesting study supporting an argument against random movement simulations recently appeared [5]. In the following we briefly report the main techniques currently proposed for LBS privacy preservation, identifying the ones similar to those tested in our experiments, and the ones using simulations to generate the datasets for experiments.

Privacy preserving solutions based on cryptographic techniques that totally hide the location information in requests, even to the SP, have been recently proposed [6] for LBS based on 1-NN queries, and may be probably adapted for the service we consider. If proven to be correct, no simulation would be needed for these techniques since no information would leak from any request and the above privacy threats do not apply. However, this adaptation is still to be investigated, and there are some general concerns with these approaches regarding efficiency and flexibility.

A popular alternative technique is spatial cloaking, consisting in the generalization of the spatial information transmitted to the SP as part of a service request. By receiving generalized locations, the SP can only return approximate results on the presence of close-by group members and their positions; while it may be possible to have a trusted entity in the middle filtering the communication and improving the precision, the related overhead costs should be taken into account in evaluating the trade-off between generalization and quality of service. While in this paper we consider techniques based on spatial cloaking as in [7–9, 2, 10], other proposals have considered different techniques, including the generation of dummy requests, the use of incremental requests, or the substitution in the request of the position of the issuer with a region that does not include her (see among others [11–13]).

Most of the proposals for LBS privacy have only considered requests in isolation while a few have also addressed the cases in which sequences of requests can be exploited by the adversary ([14, 15] among others), as in the friend-finder service. A related problem is privacy-aware publication of trajectories [16, 17]; even if this has some aspects more similar to database publication than to service request privacy preservation, we believe that our results may be important for these studies as well.

Synthetic, mostly random, user movements obtained by the Brinkhoff simulator or other simulators have been used in most of the above cited papers as well as in our own previous work.

Organization

The rest of the paper is organized as follows. In Section 2 we formally define how we evaluate the privacy of LBS requests, or equivalently, how we measure the risk of a privacy violation upon issuing a request. In Section 3 we explain how the two datasets were obtained from the generators based on the parameters characterizing the deployment scenario. In Section 4 we briefly explain the privacy preservation algorithms being used and we report our experimental results. Section 5 concludes the paper.

2 Privacy metric of generalized requests

As mentioned in the introduction, we are concerned with privacy protection via location generalization (also called spatial cloaking). In this section, we formalize the adversary model we consider in this paper, and give a metric to measure the privacy provided by a set of generalized requests against the adversary.

2.1 Requests, original requests, and generalized requests

We first formally define requests and generalized request for LBS. A request issued by a user without alteration is called an *original* request, and a *generalized* request is one that is sent to the service provider and has been altered from the original one for the purpose of privacy protection. Both kinds of requests are called *requests* and denoted r. A convention in this paper is to use r' to denote generalized requests to emphasize its generalized nature, while use r to denote original requests, if not specified otherwise.

Either the client software or a trusted medium transforms (or *generalizes*) an original request to a generalized one. In this paper, we are not concerned about *how* the generalization has happened, but rather on the *resulting* generalized requests and their privacy properties. In the experimental section, we evaluate the performance of generalization algorithms based on the generalized requests they generate.

Each LBS *request* r, either original or generalized, is logically divided into three parts: $IDdata$, $STdata$, and $SSdata$, containing user identification data, location and time of the request, and other service parameters, respectively. In the sequel, the spatial and temporal components in $STdata$ are denoted with $Sdata$ and $Tdata$, respectively. In this paper, for the sake of simplicity, we consider space and time as discrete domains. However, our results can be easily extended to the case in which these two domains are continuous.

Each generalized request r' must correspond to an original request r such that the difference between r and r' is only in $SData$ and furthermore, $r.Sdata \subseteq r'.Sdata$, i.e., the spatial region of the generalized request must contain (or be equal to) the spatial region of the original request[4]. We use $issuer(r)$ to denote the actual issuer of the (original or generalized) request r.

[4] Here "region" can be a point.

2.2 Adversary model

The objective here is to provide an adversary model that captures a general class of adversary models. In a sense, our adversary model is an adversary "meta-model". This adversary meta-model concerns two aspects of knowledge that an adversary might have: (1) knowledge of users' whereabouts (i.e., their locations), and (2) correlation of (generalized) requests. These two aspects cover the (explicit or implicit) assumptions appeared in the relevant literature.

For users' locations, we assume that the adversary has the knowledge expressed as the following *Ident* function:

$$Ident_t : the\ Areas \longrightarrow the\ User\ sets,$$

that is, given an area A, $Ident_t(A)$ is the set of users whom, through certain means, the adversary has identified to be located in area A at time t. In the following, when no confusion arises, we omit the time instant t. We further assume that this knowledge is *correct* in the sense that these identified users in reality are indeed in area A at the time.

For a given user i, if there exists an area A such that $i \in Ident(A)$, then we say i is *identified* by the adversary. Furthermore, we say that i is *identified in* A. Note that there may be users who are also in A but the adversary does not identify them. This may happen either because the adversary is not aware of the presence of users in A, or because the adversary cannot identify these users even if he is aware of their presence. We do not distinguish these two cases in our adversary model as we shall see later that the distinction of the two cases does not make any perceptible difference in the ability of the adversary when the total population is large.

Clearly, in reality, there are lots of different sources of external information that can lead the adversary to estimate the location of users. Some may lead the adversary to know that a user is in a certain area, but not the exact location. For example, an adversary may know that Bob is in a pub (due to his use of a fidelity card at the pub), but may not know which room he is in. Some statistical analysis may be done to derive the *probability* that Bob is in a particular room, but this is beyond the scope of this paper.

The most conservative assumption regarding this capability of the adversary is that $Ident(A)$ will give *exactly* all the users for each area A. It can be seen that if the privacy of the user is guaranteed in this most conservative assumption, then privacy is also guaranteed against any less precise $Ident$ function. However, this conservative assumption is unlikely true in reality, while some observed that this assumption degenerates the quality of service unnecessarily. It will be interesting to see how much privacy and quality of service change with more realistic $Ident$ functions. This is partly the goal of our paper.

As part of this adversary model regarding the location and users, we also assume another function:

$$Num_t : the\ Areas \longrightarrow [0, \infty),$$

that is, given an area A, $Num_t(A)$ gives an estimate of the number of users in the area at time t. This function can be derived from statistical information publicly available or through some kind of counting mechanism such as tickets to a theater. Again, when no confusion arises, we do not indicate the time instant t.

The second part of the adversary model is his ability to correlate requests. We formalize this with the following function L:

$$L : the\ Requests \longrightarrow the\ Request\ sets,$$

that is, given a (generalized) request r', $L(r')$ gives a set of requests such that the adversary has concluded, through certain means, are issued by the same user who issued the request r'. In other words, all the requests in $L(r')$ are *linked* to r', although the adversary may still not know who the user is.

Note that $L(r)$ may only give an (often small) subset of all the requests issued by the issuer of r. On the other hand, we assume that the function L is *correct* in the sense that each request in $L(r)$ is indeed issued by the same user in reality. A set of requests is called a *trace*, denoted τ, if from the link function L we understand that all requests are issued by the same user. The requests in τ are implicitly ordered along the time dimension.

As in the case for $Ident$ function, the most conservative assumption on correlation is that $L(r)$ gives exactly *all* the (generalized) requests that are issued by the issuer of r. This is a very strong assumption that may lead to severely decrease quality of service when accompanied with the most conservative assumption about the $Ident$ function. Again, a partial goal of this paper is to study the impact of a less conservative but more realistic assumption on L.

In [2], we proposed a formal framework to model LBS privacy attacks and defenses for the static case. The main idea is that the safety of a defense technique can be formally evaluated only if the *context*, i.e., the assumptions about the adversary's external knowledge, is explicitly stated. Following this methodology, in this paper, a context C_H is given by three functions $Ident$, Num, and L, that is

$$C_H = (Ident, Num, L).$$

In the next section, we formalize the attack on the generalized requests that an adversary can perform in a context C_H.

A consequence of restricting to context C_H is that, analogously to the related work in this area, we focus our attention on using only *STdata* as a *quasi-identifier*. Intuitively, a quasi-identifier in a request is a combination of values that can be used to provide more information on who the actual issuer of a request may be than without these values. For example, if the $Ident$ function is given, the *STData* in the request is a quasi-identifier as it may provide information on the actual issuer, as shown in the next subsection. In principle, any information contained in a request should be carefully analyzed to see if it may serve as a quasi-identifier. For example, the IDdata part is an obvious target, and some service specific parameters may be used to link the request to users. However, these aspects are outside the scope of this paper.

2.3 Privacy Evaluation

The general question for this subsection is, given a set of generalized requests and a context C_H, how much privacy the users who issued these requests have.

We want to find the following function:

$$Att : the\ Request\ set \times the\ Users \longrightarrow [0, 1],$$

Intuitively, given a (generalized) request r' and a user i, $Att(r', i)$ gives the probability that the adversary can derive from C_H that i is the issuer of r' among all the users.

In the following of this section we show how to specify the attack function for context C_H. Once the attack function is specified, we can use the following formula to evaluate the privacy value of a request:

$$Privacy(r') = 1 - Att_{C_H}(r', issuer(r')) \tag{1}$$

Intuitively, this value is the probability that the attacker will not associate the issuer of request r' to r'.

In order to specify the *Att* function, we introduce the function $Inside(i, r')$ that indicates the probability of user i to be located in $r'.Sdata$ at the time of the request. Intuitively, $Inside(i, r') = 1$ if user i is identified by the adversary as one of the users that are located in $r'.Sdata$ at time $r'.Tdata$, i.e., $i \in Ident_t(r'.Sdata)$ when $t = r'.Tdata$. On the contrary, $Inside(i, r') = 0$ if i is recognized by the adversary as one of the users located outside $r'.Sdata$ at time $r'.Tdata$, i.e., there exists an area A with $A \cap r'.Sdata = \emptyset$ such that $i \in Ident(A)$. Finally, if neither of the above cases hold, then the adversary does not know where i is. There is still a probability that i is in $r'.Sdata$. Theoretically, this probability is the number of users in $r'.Sdata$ that are not recognized by the adversary (i.e., $Num(r'.Sdata) - |Ident(r'.Sdata)|$) divided by all the users who are not recognized by the adversary anywhere (i.e., $|I| - |Ident(\Omega)|$, where I is the set of all users, and Ω is the entire area for the application). Formally,

$$Inside(i, r') = \begin{cases} 1 & if\ i \in Ident(r'.Sdata) \\ 0 & if\ \exists A : A \cap r'.Sdata = \emptyset\ and\ i \in Ident(A) \\ \frac{Num(r'.Sdata) - |Ident(r'.Sdata)|}{|I| - |Ident(\Omega)|} & otherwise \end{cases} \tag{2}$$

We can now define the *Att* function in context C_H. For the sake of presentation, let us first consider the attack in the snapshot context

$$C_{snap} = (Ident, Num, L_{snap}),$$

where for each generalized request r', $L_{snap}(r') = \{r'\}$. In this special case, the probability of a user i of being the issuer of r' is given by the probability of i being in $r'.Sdata$ at the time of the request, normalized among all the users in I. Formally, the attack can be defined as:

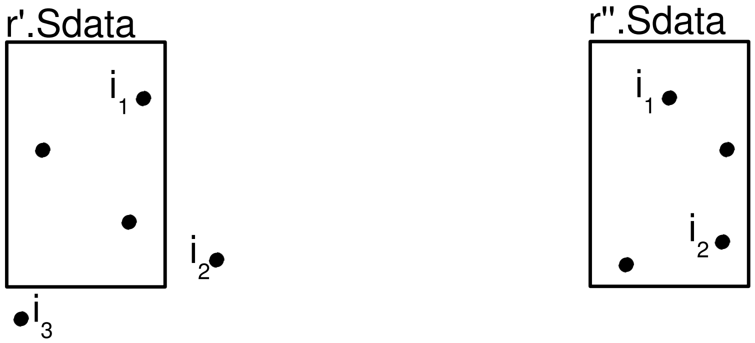
$$Att_{C_{snap}}(r', i) = \frac{Inside(i, r')}{\sum_{i' \in I} Inside(i', r')} \tag{3}$$

When the total population of users is large (relative to the number of users whose locations are known to the adversary), then the "otherwise" case in Formula 2 is very small, albeit nonzero. Intuitively, if a user i falls into this case, then the adversary cannot really distinguish this particular user from all other users who also fall into this case. For such a user i, we can simply give a value $1/|I|$ to $Att_{C_{snap}}(r', i)$. We could give $1/(|I| - Num(\Omega))$, but this does not make much impact in practice. Now it's easy to see that

$$Att_{C_{snap}}(r', i) \approx \begin{cases} 1/Num(r'.Sdata) & \textit{if } Inside(i, r') = 1 \\ 0 & \textit{if } Inside(i, r') = 0 \\ 1/|I| & \textit{otherwise} \end{cases} \tag{4}$$

The above formula makes intuitively sense. Indeed, if i is recognized as inside $r'.Sdata$, without any other information, the adversary cannot distinguish him/her from any of the $Num(r'.Sdata)$ people in the area who might be the issuer. If i is recognized outside, then clearly i cannot be the issuer due to our definition of (generalized) requests. If i is not recognized anywhere (meaning he/she can be anywhere), then the attacker cannot distinguish him/her from any of the other people who are not recognized. Since we assume the total population is much greater than $Num(\Omega)$, the probability that i is the issuer is close to $1/|I|$.

Example 1. Consider the situation shown in Figure 1(a) in which there is the request r' such that, at time $r'.Tdata$, there are three users in $r'.Sdata$: one of them is identified as i_1, the other two are not identified. The adversary can also identify users i_2 and i_3 outside $r'.Sdata$ at time $r'.Tdata$. Assume that the set I contains 100 users.



(a) First request, r'. (b) Second request, r''.

Fig. 1. Example of attack

Clearly, i_2 and i_3 have zero probability of being the issuers, since they are identified outside $r'.Sdata$ and due to the assumption that the spatial region of any generalized request must contain the spatial region of the original request. On the contrary, the adversary is sure about the fact that i_1 is located in $r'.Sdata$. By Equation 3, the attack associates i_1 to r' with likelihood $1/(\sum_{i' \in I} Inside(i', r'))$. By Formula 2, for each user i in $I \setminus \{i_1, i_2, i_3\}$, $Inside(i, r') = 2/100$. Therefore, $\sum_{i' \in I} Inside(i', r') = 97 * 2/100 + 1 \approx 3$. Consequently, the probability of i_1 to be the issuer of r' is approximately 1/3. Moreover, each user $i \in I \setminus \{i_1, i_2, i_3\}$ has a probability to be the issuer of about $(2/100)/3 = 2/300$.

In the general case $L(r') \supseteq \{r'\}$, we can evaluate, analogously to the snapshot case, the probability that a user is located in the generalized region of all the requests in the trace $\tau = L(r')$. So, we can extend the *Inside* function to traces where, given a trace τ and a user i, $Inside(i, \tau)$ is the probability that user i is located, for each request r' in τ, in $r'.STdata$. Then, the attack is

$$Att_{C_H}(r', i) = \frac{Inside(i, L(r'))}{\sum_{i' \in I} Inside(i', L(r'))} \tag{5}$$

We now turn to consider how to compute $Inside(i, \tau)$.

First consider some easy cases. If $i \in Ident(r')$ for all requests $r' \in \tau$, then $Inside(i, \tau) = 1$. If $i \in Ident(A)$ and $A \cap r'.Sdata = \emptyset$ for an area A and at least one requests $r' \in \tau$, then $Inside(i, \tau) = 0$.

The rest of cases are difficult ones. To calculate $Inside(i, \tau)$, we need to consider the likelihood of someone moving from one location to/from another in the specific times. In this paper, we advocate the following as a reasonable approach. We assume for each pair of locations A and B and two times t_1 and t_2, we know the probability of a user i being in B at time t_2 conditioned on the fact that the user is in A at time t_1. In formalism, consider two random variables X: "i is inside A at time t_1" and Y: "i is inside B at time t_2", where A and B are two areas and t_1 and t_2 are two different times. We assume the adversary knows the value $P(Y|X)$.

We note that $P(Y|X)$ in general is not the same as $P(Y)$. Indeed, how likely user i is in B can depend on how likely the same user is in A. Take two extreme examples: if A and B are very far away and t_1 and t_2 are close to each other, then i cannot be in B at t_2 if i is in A at t_1, i.e., $P(Y|X) \approx 0$. On the other hand, if A and B are just two locations along a one-way road and the difference between times t_1 and t_2 matches the time needed to move from A to B with a normal moving speed, then $P(Y|X) \approx 1$. In practice, this value can be derived from historical observations and experiences.

Now, assume τ consists of the requests $r'_1, \ldots, r'_k$. We form a Bayesian network for each user i with X_1, …, X_k as the nodes, where each X_j corresponds to the random variable: "user i is in $r_j.Sdata$ at time $r_j.Tdata$". In this network, for each node X_h that satisfies the condition (denoted c) $i \in Ident_t(r'_h.Sdata)$ with $t = r'_h.Tdata$, we draw an arc towards each other node X'_h which does not satisfy condition C. In addition, for each pair of nodes r'_h and r''_h such that neither satisfy condition c, we draw an arc from X'_h to X''_h if the $r'_{h'}.Tdata < r'_{h''}.Tdata$. (The resulting network is acyclic.) As we have assumed, we know the value $P(X'_h|X_h)$ for each arc X_h to X'_h. Denote by E the conjunctive fact that $P(X_h) = 1$ for each $r_h \in \tau$ that satisfies condition c. What we want to find is $P(X_1, \ldots, X_k|E) = Inside(i, \tau)$. This is a well-studied belief revision problem, and many computation and approximation methods exist. (Note that if we apply this method to the easier cases mentioned earlier, we would arrive at the correct values.)

Example 2. Continue from Example 1 and assume a second request r'' (see Figure 1(b)) is issued after r' and that r'' is linked with r', so τ consists of these two requests. At time $r''.Tdata$, there are 4 users inside $r''.Sdata$, two of which are identified as i_1 and i_2. No user is identified outside $r''.Sdata$. From the above discussion, it follows that $Inside(i_2, \tau) = Inside(i_3, \tau) = 0$ since i_2 and i_3 are identified outside the first generalized request r'. All the other users have a non-zero probability of being inside the generalized region of each request in the trace. In particular, $Inside(i_1, \tau) = 1$ since i_1 is recognized in both requests. Consider a user $i \in I \setminus \{i_1, i_2, i_3\}$, and denote X and Y being the assertion that "i is in $r'.Sdata$ at time $r'.Tdata$" and "i is in $r''.Sdata$ at time $r''.Tdata$". In this case, the Bayesian network for i has two nodes $X_{r'}$ and $X_{r''}$, and there is an arc from $X_{r'}$ to $X_{r''}$ since r'' is issued after r' is. Now let us assume $P(X_{r''}|X_{r'}) = 0.75$, i.e., there is a 75% likelihood that someone in $r'.Sdata$ will move to $r''.Sdata$ at the specified times. Now compute $Inside(i, \tau) = P(X_{r'}, X_{r''}) = P(X_{r'})P(X_{r''}|X_{r'}) = 2/97 * 0.75$. Now the sum of all the $Inside(j, \tau)$ value is $1 + 0 + 0 + 97 * 2/97 * 0.75 = 2.5$. The attack value under these assumptions then is as follows: For $Att_{C_H}(r'', i_1) = 1/2.5 = 40\%$, $Att_{C_H}(r'', i_2) = Att_{C_H}(r'', i_3) = 0$, and $Att_{C_H}(r'', i) = (2/97) * .75/2.5 \approx 0.6\%$ for all other 97 users i.

To make the situation more interesting, let us remove the fact that i_2 was recognized outside at time $r'.Tdata$, and we want to figure out the value $Inside(i_2, \tau)$. In this case, let us assume $P(X_{r'}|X_{r''}) = 0.75$, namely people who are in $r''.Sdata$ have a 75% likelihood to be from $r'.Sdata$. Under the fact E that i_2 is in $r''.Sdata$, then we know $Inside(i_2, \tau) = P(X_{r'}, X_{r''}|E) = 0.75$. Then the sum of $Inside$ values is $1+0.75+0+97*2/97*0.75 = 3.25$. Hence, $Att_{C_H}(r'', i_1) \approx 31\%$, $Att_{C_H}(r'', i_2) \approx 23\%$, $Att_{C_H}(r'', i_3) = 0$, and $Att_{C_H}(r'', i) = (2/97)*0.75/3.25 \approx 0.47\%$ for each other 97 users i. This is an interesting exercise as it reveals that if we add i_2 to be possibly in $r'.Sdata$ (with 75% probability), then the likelihood that i_1 is the issuer decreases, which is intuitively correct.

It is worth noting that the definition of attack in context C_H is a proper extension of the attack that can be defined in the conservative context in which the adversary knows the location and the identity of each user in each time instant. The historical attack in this context was first proposed in [14]. The idea is that the only users that have non-zero probability of being the issuer of a trace of requests are those whose spatio-temporal location is contained in the generalized region of every request in the trace. It can be easily seen that, if each user can be identified at each time instant, then the $Inside()$ function returns either 0 or 1 and hence the attack we specified for context C_H assigns a zero probability to each user that is located outside the generalized region of any request in the trace.

3 The *MilanoByNight* simulation

In order to evaluate privacy-preserving techniques applied to LBS, a dataset of users' movements is needed. In our experiments, we want to focus on privacy threats that arise when using a friend finder service, as described in Section 1. We suppose that this kind of service is primarily used by people during entertainment hours, especially at night. Therefore, the ideal dataset for our experiments should represent movements of people on a typical Friday or Saturday night in a big city, when users tend to move to entertainment places. To our knowledge, currently there are no datasets like this publicly available, specially considering that we want to have large scale, individual, and precise location data (i.e., with the same approximation of current consumer GPS technology). In this section we describe how we generated this user movement dataset.

3.1 Relevant Parameters

For our experiments we want to artificially generate movements for $100,000$ users on the road network of Milan[5]. The total area of the map is 324 km^2, and the resulting average density is 308 users/km^2. Very detailed digital vector maps of the city have been generously provided by the municipality of Milan. The simulation includes a total of $30,000$ home buildings and $1,000$ entertainment places; the first value is strictly related to the considered number of users, while the second is based on real data from public sources which also provide the geographical distribution of the places. Our simulation starts at 7 pm and ends at 1 am. During these hours, each user moves from house to an entertainment place, spends some time in that place, and possibly moves to another entertainment place or go back home.

All probabilities related to users' choices are modeled with a probability distributions. For this specific data generation, some of the important parameters of the simulation are:

- **Source and destination**. These are the locations essential to define movements. They may be homes or entertainment places. Some places in some districts are more popular than others.
- **StartingTime**. The time at which a user leaves her home to go to the first entertainment place.
- **Permanence**. How long will a user stay at one entertainment place?
- **NumPlaces**. How many entertainment places will a user visit on one night?

In order to have a realistic model of these distributions, we prepared a survey to collect real users data. We are still collecting data, but the current parameters are based on interviews of more than 300 people in our target category.

[5] $100,000$ is an estimation of the number of people participating in the service we consider.

3.2 Weaknesses of mostly random movement simulations

Many papers in the field of privacy preservation in LBS use artificial data generated by moving object simulators to evaluate their techniques. However, most of the simulators are usually not able to reproduce a realistic behavior of users. For example, objects generated by the Brinkhoff generator [3] cannot be aggregated in certain places (e.g., entertainment places). Indeed, once an object is instantiated, the generator chooses a random destination point on the map; after reaching the destination, the object disappears from the dataset. For the same reason, it is not possible to reproduce simple movement patterns (e.g.: a user going out from her home to another place and then coming back home), nor to simulate that a user remains for a certain time in a place.

Despite these strong limitations, we made our best effort to use the Brinkhoff simulator to generate a set of user movements with characteristics as close as possible to those explained in Section 3.1. For example, in order to simulate entertainment places, some random points on the map, among those points on the trajectories of users, were picked. The simulation has the main purpose of understanding if testing privacy preservation over random movement simulations gives significantly different results with respect to more realistic simulations.

3.3 Generation of user movements with a context simulator

In order to obtain a dataset consistent with the parameters specified in Section 3.1, we need a more sophisticated simulator. For our experiments, we have chosen to customize the Siafu context simulator [4]. With a context simulator it is possible to design models for agents, places and context. Therefore, it is possible to define particular places of aggregation and make users dynamically choose which place to reach and how long to stay in that place. In our simulation homes are distributed almost uniformly on the map, with a minor concentration on the central zones of the city. Entertainment places are mostly concentrated in 5 zones of the city.

The distributions for *StartingTime*, *Permanence* and *NumPlaces* parameters introduced in Section 3.1 were modeled with the results of the survey. For example, the time of permanence in an entertainment place was modeled according to the following percentages derived from the survey: 9.17% of the users stays less than 1 hour, 34.20% stays between 1 and 2 hours, 32.92% stays between 2 and 3 hours, 16.04% stays between 3 and 4 hours, and 7.68% stays more than 4 hours.

Following these parameters, in our dataset users spend 50.87% of the time at home, 7.28% of the time moving from one place to another and 41.85% of the time in entertainment places. When a user moves from one place to another, she decides whether to go on foot or by car. In general, if an entertainment place is farther than 500 meters, people tend to move by car, and this is reflected in the simulation. The average speed of movements by car is 20 km/h, while the average speed on foot is 3.6 km/h. With our parameters 10.64% of movements are done on foot, while all the others are done by car.

4 Experimental results

In this section we show the results of our experimental evaluation. We first define how we evaluate the quality of service in Section 4.1, then we describe the experimental setting in Section 4.2 and the generalization algorithms we used in Section 4.3. Finally, in Sections 4.4 and 4.5 we show the impact of the simulation parameters and of the user movements, respectively, in the evaluation of the generalization algorithms.

4.1 Evaluation of the Quality of Service

Different metrics can be defined to measure QoS for different kind of services. For instance, for the friend-finder service we are considering, it would be possible to measure how many times the generalization leads the SP to return an incorrect result i.e., the issuer is not notified of a close-by friend or, vice versa, the issuer is notified for a friend that is not close-by. While this metric is useful for this specific application, we want to measure the QoS independently from the specific kind of service. For this reason, in this paper we evaluate how QoS degrades in terms of the perimeter of the generalized region. If the generalized region is too large, the service becomes useless. For this purpose, we introduce a new parameter, called $maxP$, that indicates this threshold in terms of the maximum perimeter. We assume that no request is sent to the SP with a perimeter larger than $maxP$.

4.2 Experimental settings

In our experiments we used two datasets of users movements. The dataset AB (Agent-Based) was generated with the customized Siafu simulator as described in Section 3.3, while the dataset MRM (Mostly Random Movement) was created with the Brinkhoff simulator as described in Section 3.2. In both cases, we simulate LBS requests for the friend-finder service by choosing random users in the simulation, we compute for each request the generalization according to a given algorithm, we evaluate QoS as explained in Section 4.1 and privacy according to formula (1) presented in Section 2.

The most important parameters that characterize the simulations are reported in Table 1, with the values in bold denoting default values. The *number of users* indicates how many users are in the simulation, and the simulations are designed so that this number remains almost constant at each time instant. In every two minutes, each user has a probability P_{req} of issuing a request. For technical reasons, the reported tests are based on a time frame of three hours over the total six hours of the MilanoByNight scenario. This implies that in the default case we consider a total of 45 requests (one every four minutes of the considered time frame). The parameter P_{id-in} indicates the probability that a user is identified when she is located in a entertainment place while P_{id-out} is the probability that a user is identified in any other location (e.g., while moving from home to a entertainment place). While we also perform experiments where the two probabilities are the same, our scenario suggests as much more realistic a higher value for P_{id-in} (it is considered ten times higher than P_{id-out}). This is due to the fact that restaurants, pubs, movie theaters, and similar places are likely to have different ways to identify people (fidelity or membership cards, wifi hotspots, cameras, credit card payments, etc.) and in several cases more than one place is owned by the same company that may have an interest in collecting data about its customers.

Finally, P_{link} indicates the probability that two consecutive requests can be identified as issued by the same user.[6] While we perform our tests considering a full range of values, the specific default value reported in the table is due to a recent study on the ability of linking positions based on spatio-temporal correlation [18].

Table 1. Parameter values

Parameter	**Values**
dataset	***AB***, *MRM*
number of users	10k, 20k, 30k, 40k, 50k, 60k, 70k, 80k, 90k, **100k**
P_{req}	0.1, 0.2, 0.3, 0.4, **0.5**, 0.6, 0.7, 0.8, 0.9, 1.0
P_{id-in}	0.1, **0.2**, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0
P_{id-out}	0.01, **0.02**, 0.03, 0.04, 0.05, 0.06, 0.07, 0.08, 0.09, 0.1
P_{link}	0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, **0.87**, 0.9, 1.0

The experimental results we show in this section are obtained by running the simulation for 100 issuers and then computing the average values.

4.3 The Generalization Algorithms Used in the Experiments

In our experiments we evaluate the privacy and the QoS of requests generalized by using two algorithms previously proposed in the literature. The first one, called *Grid*, was presented in [19], and it is used as a representative of several algorithms aimed at guaranteeing k-anonymity in the snapshot case, i.e., these algorithms do not take into account link ability of the adversary. Intuitively, this particular algorithm partitions all users into blocks, each one having at least cardinality k. Then, it computes the generalized region as the minimum bounding rectangle (MBR) that covers the location of the users in the same block as the issuer.

The second algorithm, *Greedy*, was first proposed in [14] and a similar idea was also described in [15]. The use of Greedy is intended to represent algorithms aimed at preserving privacy in the historical case, i.e., the general C_H context,

[6] The limitation to consecutive requests is because in our specific scenario we assume linking is performed mainly through spatio-temporal correlation.

assuming that the attacker may actually obtain and recognize traces of requests from the same issuer. This algorithm computes the generalization of the first request r in a trace using an algorithm for the snapshot case. While doing this, the set A of users located in the generalized region is stored. The generalized regions of the successive request r' linked with r is then computed as the MBR of the location of the users in A at the time of r'. In our implementation we use *Grid* as the snapshot algorithm to compute the generalization of the first request.

For the purpose of our tests, we modified the two algorithms above so that each generalized region has a perimeter always smaller than $maxP$. To achieve this, if the perimeter of the generalized region is larger than $maxP$, then the region is iteratively shrunk, until its perimeter is below $maxP$, by excluding from the MBR the user that is farther from the issuer. In the *Greedy* algorithm, when a user is excluded from the generalized region, then it is also excluded from the set A of users, and hence he is not used in the generalization of the successive requests. Eventually, when the set A contains the issuer only, a snapshot generalization is executed again and A is reinitialized.

In addition to the input request r, and the location of all the users in the system, the considered algorithms require two additional parameters: the value k, and the threshold $maxP$. In our tests, we used values for k between 10 and 60 (default is 10) and values for $maxP$ between 1000 to 4000 meters (default is 1000 meters).

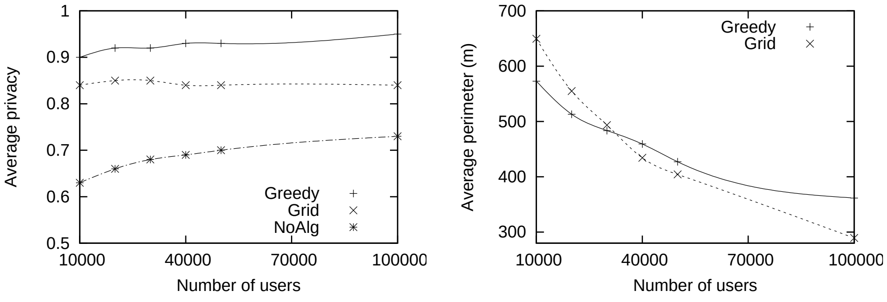
In our experimental results we also evaluated the privacy threat when no privacy preserving algorithm is applied. The label *NoAlg* is used in the figures to identify results in this particular case.

4.4 Impact of Simulation Parameters in the Evaluation of the Generalization Algorithms

The objective of the first set of experimental results we present is to show which parameters of the simulation affect most the evaluation of the generalization algorithms. In these tests we used the *AB* dataset only.

Figure 2(a) shows that the average privacy obtained with *Greedy* and *Grid* is not significantly affected by the size of the total population. Indeed, both algorithms, independently from the total number of users, try to have generalized regions that cover the location of k users, so the privacy of the requests is not affected. However, when the density is high, the two algorithms can generalize to a small area, while when the density is low, a larger area is necessary to cover the location of k users (see Figure 2(b)). On the contrary, the privacy obtained when no generalization is performed is significantly affected by the total population. Indeed, a higher density increases the probability of different users to be in the same location and hence it increases privacy also if the requests are not generalized.

A parameter that significantly affects the average privacy is the probability of identification of a user in a certain place. In Figure 3 we show the experimental results for different values of P_{id-in} when, in each test, P_{id-out} is set to



(a) Average privacy.

(b) Average perimeter.

Fig. 2. Performance evaluation for different values of the total population.

$P_{id-in}/10$. As expected, considering a trace of requests, the higher is the probability of identifying users in one or more of the regions from which the requests in the trace were performed, the smaller is the level of privacy.

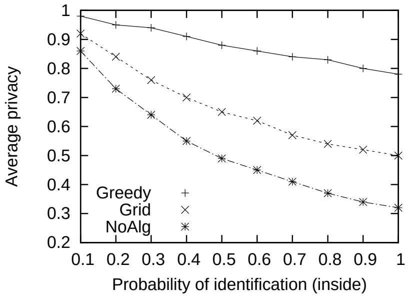
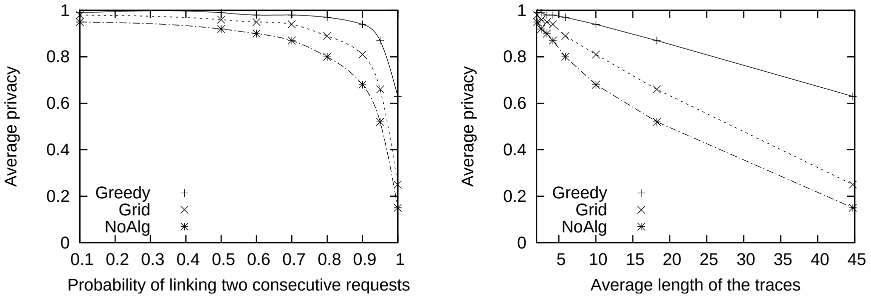


Fig. 3. Average privacy for different values of P_{id-in} ($P_{id-out} = P_{id-in}/10$).

Figure 4(a) shows the impact of P_{link} on the average privacy. As expected, high values of P_{link} lead to small values of privacy. Our results show that the relation between the P_{link} and privacy is not linear. Indeed, privacy depends almost linearly on the average length of the traces identified by the adversary (Figure 4(b)). However, the average length of the traces grows almost exponentially with the value of P_{link} (Figure 5).

To summarize the first set of experiments, our findings show that many parameters of the simulation significantly affect the evaluation of the generalization algorithms. This implies that when a generalization algorithm is evaluated it is necessary to carefully estimate realistic values for the parameters of the simulation. Indeed, an error in the estimation may lead to misleading results.



(a) Average privacy as a function of P_{link}. (b) Average privacy as a function of the average trace length.

Fig. 4. Performance evaluation for different values of P_{link}.

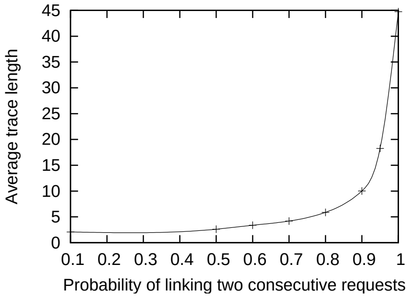


Fig. 5. Average trace length as a function of P_{link}.

4.5 Impact of the User Movements on the Evaluation of the Generalization Algorithms

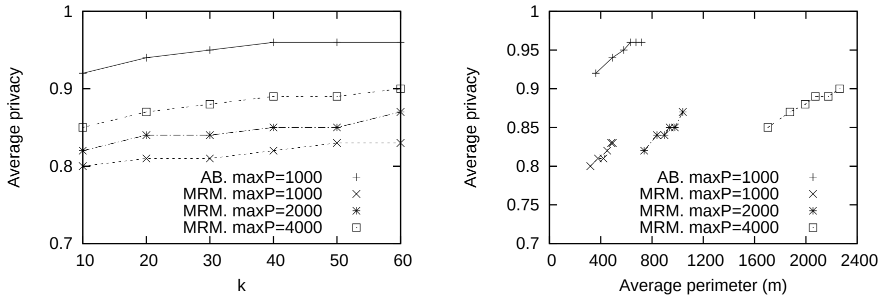
The objective of the second set of experiments is to answer an important question posed in this paper: what is the impact of the different simulated user movements on the evaluation of the Generalization Algorithms? We answer to this question with a set of tests performed on the two different datasets we obtained as described above.

The first set of tests, reported in in Figure 6, compares the privacy achieved by the Greedy algorithm on the two datasets for different values of k and for different values of QoS. The experiments on MRM were repeated trying also larger values for the QoS threshold ($maxP = 2000$ and $maxP = 4000$), so three different versions of MRM appear in the figures. In order to focus on these parameters only, in these tests the probability of identification was set to the same value for any place ($P_{id-in} = P_{id-out} = 0.1$), and for the MRM dataset the issuer of the requests was randomly chosen only among those that stay in the simulation for 3 hours, ignoring the ones staying for much shorter time that inevitably are part of this dataset. This setting allowed us to compare the results on the two datasets using the same average length of traces identified by the adversary.

Figure 6(a) shows that the average privacy of the algorithm evaluated on the AB dataset is much higher than on the MRM dataset. This is mainly motivated by the fact that in AB users tend to concentrate in a few locations (the entertainment places) and this enhances privacy. This is also confirmed by a similar test performed without using any generalization of locations; we obtained values constantly higher for the AB dataset (the average privacy is 0.67 in AB and 0.55 in MRM).

In Figure 6(b) we show the QoS achieved by the algorithm in the two datasets with respect to the average privacy achieved. This result confirms that the level of privacy evaluated on the AB dataset using small values of k and $maxP$ for the algorithm cannot be observed on the MRM dataset even with much higher values for these parameters.

From the experiments shown in Figure 6 we can conclude that if the MRM dataset is used as a benchmark to estimate the values of k and $maxP$ that are necessary to provide a desired average level of privacy, then the results will suggest the use of values that are over-protective. As a consequence, it is possible that the service will exhibit a much lower QoS than the one that could be achieved with the same algorithm.



(a) Average privacy as a function of k.

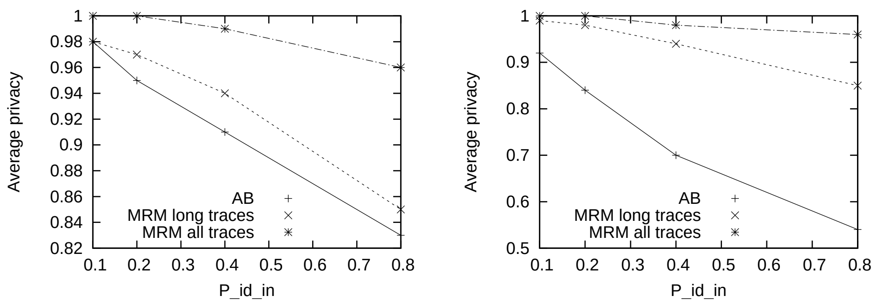
(b) Average privacy as a function of the average perimeter.

Fig. 6. Evaluation of the *Greedy* algorithm using AB and MRM data sets. $P_{id-in} = P_{id-out} = 0.1$

The above results may still support the safety of using MRM, since according to what we have seen above a technique achieving a certain level of privacy may only do better in a real scenario. However, our second set of experiments shows that this is not the case.

In Figure 7 we show the results we obtained by varying the probability of identification. For this test, we considered two sets of issuers in the MRM data set. One set is composed by users that stay in the simulation for 3 hours, (MRM *long traces*, in Figure 7), while the other contains issuers randomly chosen in the entire set of users (*MRM all traces*, in Figure 7), hence including users staying in the simulation for a much shorter time.

In Figure 7(a) and 7(b) we can observe that the execution on the *MRM* dataset leads to evaluate a privacy level that is higher than the one obtained on the *AB* dataset. In particular, the evaluation of the *Grid* algorithm using the *MRM* dataset (Figure 7(b)), would suggest that the algorithm is able to provide a high privacy protection. However, when evaluating the same algorithm using the more realistic dataset *AB*, this conclusion seems to be incorrect. In this case, the evaluation on the *MRM* dataset may lead to underestimate the privacy risk, and hence to deploy services based on generalization algorithms that may not provide the minimum required level of privacy.



(a) *Greedy* algorithm.

(b) *Grid* algorithm.

Fig. 7. Average privacy using *AB* and *MRM* data sets. $P_{id-out} = P_{id-in}/10$.

5 Conclusions and open issues

In this paper we claim that the experimental evaluation of LBS privacy preserving techniques should be based on user movement datasets obtained through simulations tailored to the specific deployment scenario of the target services. Our results support our thesis for the class of LBS known as friend-finder services, for techniques based on spatial cloaking, and for adversary models that include the possibility for the adversary to occasionally recognize people in certain locations. We believe that these results can be generalized to other LBS, techniques and adversary models. For example, as a future work, it would be interesting to also evaluate some defense techniques that generalize the issuer's location to an area that does not necessarily contain the issuer's location. Moreover, in our experiments we only considered the first of the two privacy threats presented in the introduction. We do have some ideas on how to extend them to consider the second, location privacy, as well. Finally, we believe a significant effort should be devoted to the development of new flexible and efficient context-aware user movement simulators, as well as to the collection of real data, possibly even in an aggregated form, to properly tune the simulations. In our opinion this is a necessary step to have significant common benchmarks to evaluate LBS privacy preserving techniques.

Acknowledgments

The authors would like to thank Stefano Varesi for his contribution in writing the code that was used for our simulations. This work was partially supported by National Science Foundation (NSF) under grant N. CNS-0716567, and by Italian MIUR under grant InterLink II04C0EC1D.

References

1. Barkhuus, L., Dey, A.: Location-based services for mobile telephony: a study of users privacy concerns. In: Proc. of the 9th International Conference on Human-Computer Interaction, IOS Press (2003) 709–712
2. Bettini, C., Mascetti, S., Wang, X.S., Jajodia, S.: Anonymity in location-based services: towards a general framework. In: Proc. of the 8th International Conference on Mobile Data Management, IEEE Computer Society (2007)
3. Brinkhoff, T.: A framework for generating network-based moving objects. GeoInformatica **6**(2) (2002) 153–180
4. Martin, M., Nurmi, P.: A generic large scale simulator for ubiquitous computing. In: 3rd Annual International Conference on Mobile and Ubiquitous Systems: Networking & Services, IEEE Computer Society (July 2006)
5. Gonzalez, M.C., Hidalgo, C.A., Barabasi, A.L.: Understanding individual human mobility patterns. Nature **453** (June 2008) 779–782
6. Ghinita, G., Kalnis, P., Khoshgozaran, A., Shahabi, C., Tan, K.L.: Private queries in location based services: Anonymizers are not necessary. In: Proc. of SIGMOD, ACM Press (2008)
7. Gruteser, M., Grunwald, D.: Anonymous usage of location-based services through spatial and temporal cloaking. In: Proc. of the 1st International Conference on Mobile Systems, Applications and Services (MobiSys), The USENIX Association (2003)
8. Mokbel, M.F., Chow, C.Y., Aref, W.G.: The new casper: query processing for location services without compromising privacy. In: Proc. of the 32nd International Conference on Very Large Data Bases, VLDB Endowment (2006) 763–774
9. Gedik, B., Liu, L.: Protecting location privacy with personalized k-anonymity: Architecture and algorithms. IEEE Transactions on Mobile Computing **7**(1) (2008) 1–18
10. Kalnis, P., Ghinita, G., Mouratidis, K., Papadias, D.: Preventing location-based identity inference in anonymous spatial queries. IEEE Transactions on Knowledge and Data Engineering **19**(12) (2007) 1719–1733
11. Kido, H., Yanagisawa, Y., Satoh, T.: Protection of location privacy using dummies for location-based services. In: Proc. of the 21st International Conference on Data Engineering Workshops, IEEE Computer Society (2005)
12. Yiu, M.L., Jensen, C.S., Huang, X., Lu, H.: Spacetwist: Managing the trade-offs among location privacy, query performance, and query accuracy in mobile services. In: Proc. of the 24th International Conference on Data Engineering, IEEE Computer Society (2008)

13. Ardagna, C.A., Cremonini, M., Damiani, E., di Vimercati, S.D.C., Samarati, P.: Location privacy protection through obfuscation-based techniques. In: Proc. of the 21st Conference on Data and Applications Security. Volume 4602 of Lecture Notes in Computer Science., Springer (2007) 47–60
14. Bettini, C., Wang, X.S., Jajodia, S.: Protecting privacy against location-based personal identification. In: Proc. of the 2nd workshop on Secure Data Management. Volume 3674 of LNCS., Springer (2005) 185–199
15. Xu, T., Cai, Y.: Location anonymity in continuous location-based services. In: Proc. of ACM International Symposium on Advances in Geographic Information Systems, ACM Press (2007)
16. Abul, O., Bonchi, F., Nanni, M.: Never walk alone: Uncertainty for anonymity in moving objects databases. In: Proc. of the 24th International Conference on Data Engineering, IEEE Computer Society (2008)
17. Terrovitis, M., Mamoulis, N.: Privacy preservation in the publication of trajectories. In: Proc. of the 9th International Conference on Mobile Data Management, IEEE Computer Society (2008)
18. Vyahhi, N., Bakiras, S., Kalnis, P., Ghinita, G.: Tracking moving objects in anonymized trajectories. In: Proc. of 19th International Conference on Database and Expert Systems Applications, Springer (2008, to Appear)
19. Mascetti, S., Bettini, C., Freni, D., Wang, X.S.: Spatial generalization algorithms for lbs privacy preservation. Journal of Location Based Services **2**(1) (2008)

A Multi-Path Approach for k-Anonymity in Mobile Hybrid Networks

C.A. Ardagna[1] A. Stavrou[2] S. Jajodia[2] P. Samarati[1] R. Martin[2]

[1] University of Milan, Italy
[2] George Mason University, USA

Abstract. The ubiquitous proliferation of mobile devices has given rise to novel user-centric applications and services. In current mobile systems, *users* gain access to remote *service providers* over *mobile network operators* which are assumed to be trusted and not improperly use or disclose users' information. In this paper, we remove this assumption, offering privacy protection of users' requests again the prying eyes of the network operators, which we consider to be honest but curious. Furthermore, to prevent abuse of the communication privacy we provide, we elevate traffic accountability as a primary design requirement. We build on prior work on network k-anonymity and multi-path communications to provide communications' anonymity in a mobile environment. The resulting system protects users' privacy while maintaining data integrity and accountability. To verify the effectiveness of our approach and measure its overhead, we implemented a prototype of our system using WiFi-enabled devices. Our preliminary results indicate that the overall impact on the end-to-end latency is negligible, thus ensuring applicability of our solution to protect the privacy of real-time services including video streaming and voice activated services.

1 Introduction

Recent technology advancements in mobile and wireless devices have fostered the development of a new wave of on-line and mobile services. Due to their pervasive nature, these services are becoming increasingly popular and wide-spread. On the other hand, the accuracy, reliability and performance of location sensing technologies, have raised concerns about the protection of users' privacy. Today, there are no mechanisms to prevent wireless communications from being broadcasted to the neighboring devices thus disclosing private information about the location of users. The worst case scenario that analysts have foreseen as a consequence of an unrestricted and unregulated availability of mobile technologies recalls the "Big Brother" stereotype: a society where the secondary effect of mobile technologies – whose primary effect is to enable the development of innovative and valuable services – becomes a form of implicit total surveillance of individuals.

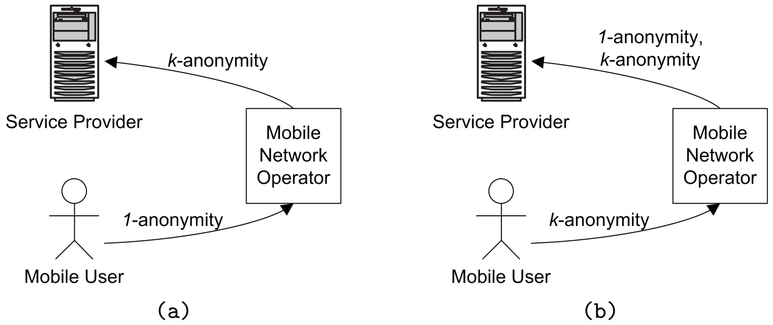


Fig. 1. Current privacy mechanisms (a) and our new vision of privacy (b)

Some recent examples can provide an idea of the extend of the problem. In September 2007, Capla Kesting Fine Art announced the plan of building a cell tower, near Brooklyn NY, able to capture, monitor and rebroadcast wireless signals, or in other terms eavesdrop WiFi communications to ensure public safety [28]. Moreover, the US Congress approved changes to the 1978 Foreign Intelligence Surveillance Act giving NSA authorization to monitor domestic phone conversations and e-mails including those stemming from the cellular network and Internet. This legislation provides the legal grounds for the cell tower's construction, and for the monitoring of users communications in the cellular network.

Current privacy protection systems are focused on preserving users from untrusted service providers. However, at the same time and assume mobile network operators to be trusted. In this paper, and to the best of our knowledge we are the first to do so, we assume mobile network operators to be honest but curious. Our approach builds on the concept of k-anonymity in the context of network communication but, unlike other approaches, aims at providing such anonymity against the mobile network operator, instead of against the service provider. Figure 1 illustrate the difference between our approach and current solutions. Current solutions (see Figure 1(a)) use k-anonymity to protect the users during the communications with the service provider and consider the mobile network operator as a fully trusted party. However, the mobile network operator has access to precise location and traffic information for each user. In our approach (see Figure 1(b)), the mobile network operator is considered honest but curious and a k-anonymity mechanism is used to protect users' privacy. The user *can* then decide if the service provider is assumed trusted. In the figure either 1-anonymity is preserved, if the service provider is assumed trusted, or k-anonymity, if the service provider is assumed untrusted. Also, our work is different from traditional research in anonymous communications [6–8, 19], because it can be applied in a mobile infrastructure and is geared towards k-anonymity, not complete sender anonymity. In addition, we treat user and traffic accountability as a fundamental requirement of our approach making sure that each user is accountable for the services requested. Having a system that can enforce data accountability prevents unwanted traffic and provides economic incentives for the deployment of privacy-preserving services.

To achieve the aforementioned goals, we extend the concept of network k-anonymity to hybrid mobile networks. In such networks, users can simultaneously create WiFi point-to-point connections, join the cellular network, and access the Internet through their mobile phones. Using a multi-path communication paradigm [23], a mobile user can achieve network k-anonymity by distributing, using WiFi network, different packets of the same message to k neighboring mobile peers, which then forward the received packet through the cellular network. This scheme achieves k-anonymity because the mobile network operator is not able to associate the users' data flow with fewer than k peers.A separate accounting mechanism can verify that the packets are legitimate. For instance, one approach is to have the data flow encrypted with a symmetric key shared between the requester and the service provider. This would assure accountability, data integrity, and confidentiality. In addition, it will prevent the abuse of anonymity [4] while providing the economic incentives to deploy anonymizing schemes. Of course, there is a clear trade-off between anonymity and latency overhead: the further we forward the packets, the better the anonymity is but the more is the latency overhead. To quantify that trade-off in practice, we have built a prototype of our system using WiFi-enabled cellphones.

The remainder of this paper is organized as follows. Section 2 illustrates the overall architecture. Section 3 discusses privacy requirements and challenges in the considered scenario and illustrates our solution. Section 4 discusses experimental results illustrating the impact of our solution on end-to-end communication. Section 5 discusses related work. Finally, Section 6 presents our conclusions.

2 Overall Architecture

Our reference model is a distributed and mobile infrastructure which forms a hybrid network [8, 9, 22], integrating both wireless, cellular and

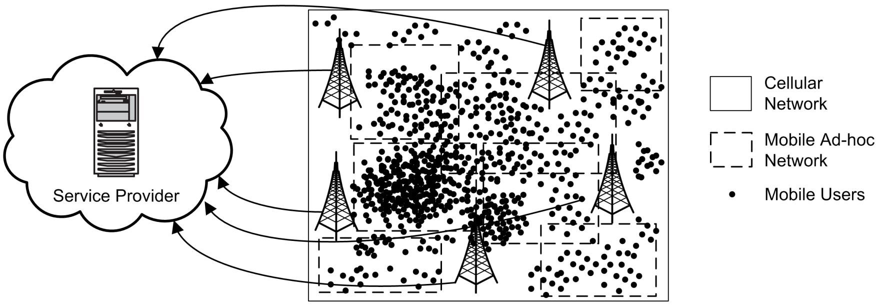


Fig. 2. Mobile Network Architecture

wired technologies. Our scenario is based on mobile parties communicating through wireless and cellular protocols to access services, either co-located in the cellular network or in the Internet. Figure 2 illustrates the overall architecture and the participating entities, which are as follows.

- *Mobile Users.* They are human users that carry mobile devices supporting both GSM/3G and WiFi protocols for communication. They request services to providers available over the network.
- *Cellular Network (and corresponding Mobile Network Operators).* It is composed of multiple radio cells (also known as cell-phone towers), which provide network access and services to mobile users. The cellular network acts as a gateway between mobile users and service providers.
- *Service Provider.* It is the entity that provides on-line services to the mobile users and collects their personal information before granting an access to its services.

Mobile users establish ad-hoc (WiFi) point-to-point connections with other mobile peers in the network, resulting in several Mobile Ad-Hoc Networks (MANETs), represented by the dashed rectangles in Figure 2. Also, mobile users receive signals from the radio cells and can connect to the cellular networks, through which they access the service. Here, we assume also mobile peers, like the provider, to be honest but curious. This means that they can try to eavesdrop a communication but do not attempt to either drop or maliciously modify it. Figure 3 illustrates the communications between the different parties in the hybrid network.

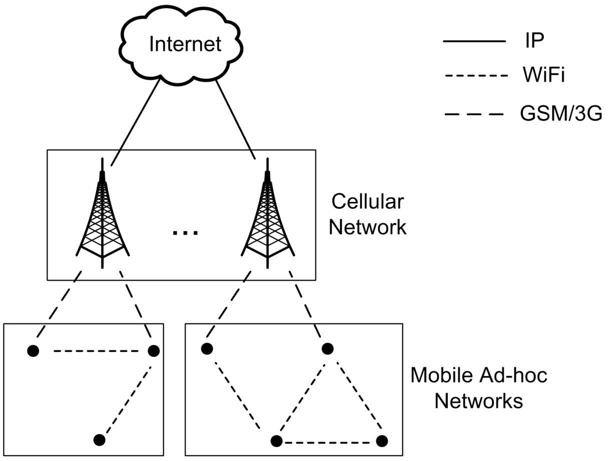


Fig. 3. Hybrid Network Communications

3 A Multi-Path Communication for Network *k*-Anonymity

We describe of our solution based on network k-anonymity by showing: *i)* how a k-anonymous request is generated and transmitted by a mobile user to the service provider through the cellular network and *ii)* how the service provider crafts a reply that can be received and decoded only by the requester concealed from the other k-1 users. Before going into details of the solution, we discuss our privacy goals and challenges.

3.1 Privacy Requirements and Challenges

In hybrid mobile networks, users privacy is at risk and is affected by several threats. In the last few years, the definition of privacy solutions was geared towards the privacy of the users, sacrificing the need for accountability. Thus, an important requirement, often neglected by mobile privacy solutions, is the necessity for mechanisms to make the users accountable for their operations. Many anonymization techniques in fact can be abused or lack economic incentives due to the lack of user accountability [4]. Service providers are often reluctant to adopt privacy solutions that completely hide the users and do not enable any form of accountability. Another challenge driving our work is the current implicit trust on mobile network operators. We believe that mobile network operators should be treated as untrusted parties with respect to confidentiality.

These challenges result in the definition of two-level privacy requirements. Two-level means that different kinds of privacy protection have to be guaranteed at: *1)* the mobile network level (*anonymous communication*) and *2)* the service level (*location hiding*).

- *Anonymous communication.* Each mobile user should communicate anonymously with the mobile network operator, possibly by masking its identity with the identities of other users joining the cellular network. At the same time, to preserve accountability, the requester's identity should be known to the service provider.
- *Location hiding.* Each mobile user interacting with a service provider should be able to hide its current location, if not otherwise required by the service provider for the service release.[3] This follows the principle of *minimum disclosure*, which states that service providers must require the least set of information needed for service provision. Conversely, location of the users must be known to the mobile network operator to provide connection to the network.

To conclude, an important requirement that any privacy solution should implement, is to provide a mechanism for expressing users' privacy preferences that strikes a balance between usability and expressiveness. In our work, the users can still express their privacy preference in terms of the number k of users that should join the *anonymity set.* This is the only effort required to the users to protect their privacy, while the application of the privacy solution is completely transparent to them.

3.2 Overview of the Approach

The concept of k-anonymity has been originally defined in the context of databases [21]. Here, we introduce a solution based on the concept of network k-anonymity, first introduced in [24], which can be defined as follow.

Definition 1 (Network k-anonymity). *Let U be a set of users and M be a message originated by a mobile user $u \in U$. User u is said to have network k-anonymity, where k is the privacy preference of the user u, if the probability of associating u as the message originator is less than or equal to $\frac{1}{k}$.*

[3] Note that, our solution is however compatible with all previous works in the context of location privacy and anonymity.

We now describe the forward and reverse anonymous communications that compose our solution. The complete protocol is shown in Figure 4. Let us define u as the mobile user that submits the request and *SP* the service provider. *SP* and the cellular network are in business relationship and u is subscribed to the cellular network. Also, *SP* and u are assumed to be in a producer-consumer relationship and to share a common secret key s that is generated through a Diffie-Hellman key exchange protocol. Each message M between a user and a service provider is encrypted thus protecting confidentiality and integrity of the message through symmetric encryption (e.g., 3DES, AES). $E_s(M)$ denotes a message M encrypted with symmetric key s. Also, a cryptographic message authentication code (i.e., $MAC_s(M)$) is calculated on the message M using s. *SP* is finally responsible for filtering of the requests.

Anonymous Request. The anonymous request process is initiated by a mobile user u, which wishes to access a service provided by service provider *SP*. No overhead is given to u in the management of the mobile and anonymous process; u needs only to specify her privacy preference k. First, $MAC_s(M)$ is calculated; then M is split in k data flows producing the set DS=$\{m_1, m_2, \ldots, m_k\}$.[4] The resulting packets are distributed among the neighbor mobile peers (peers for short) in the mobile ad-hoc network. Different algorithms, ranging from the ones based on *network state* to the ones based on *peer reputation*, can be implemented for distributing packets among peers. Here, we use a simple approach which consists in randomly forwarding the packets to the peers in u's communication range.

The distribution algorithm is illustrated in Figure 5(a) and works as follows. The requester u encrypts each packet in DS using the symmetric key s shared between u and *SP*, and then appends $MAC_s(M)$ in plaintext to each encrypted packet, that is, $E_s(DS)$ = $\{[E_s(m_1)\|MAC_s(M)], [E_s(m_2)\|MAC_s(M)], \ldots, [E_s(m_k)\|MAC_s(M)]\}$. The presence of the MAC information in every packet allows mobile peers to distinguish between packets belonging to the same message M. Requester u then randomly picks up one of the encrypted packets $[E_s(m_j)\|MAC_s(M)] \in E_s(DS)$ for sending it to the *SP*, and randomly selects $k-1$ peers in the communication range. Each selected peer receives a packet $[E_s(m_i)\|MAC_s(M)] \in E_s(DS)$ and uses a *decision forwarding function* (*dff*) to manage it. Function *dff* is defined as follow.

[4] For the sake of clarity, in the following, we use the term "packet" to identify a data flow of any dimension.

Protocol 1 *Anonymous communication protocol*

Initiator: Requester u
Involved Parties: Mobile peers *PEERS*, Mobile network operator *MNO*, Service provider *SP*
Variables: Original message M, Response message M_r, Secret key s shared between u and *SP*

```
INITIATOR ($u$)  u.1 Define message $M$ and privacy preference $k$.
                 u.2 Generate $MAC_s(M)$ and $DS = \{m_1, m_2, \dots, m_k\}$.
                 u.3 Encrypt packets in $DS$ and append $MAC_s(M)$ to them,
                     $E_s(DS) = \{[E_s(m_1) \| MAC_s(M)], \dots, [E_s(m_k) \| MAC_s(M)]\}$.
                 u.4 Select a random packet $[E_s(m_j) \| MAC_s(M)] \in E_s(DS)$.
                 u.5 Select a set of $k$-1 peers $\{p_1, \dots, p_{k-1}\} \in PEERS$.
                 u.6 Send to each $p_i \in \{p_1, \dots, p_{k-1}\}$ a packet
                     $[E_s(m_i) \| MAC_s(M)] \in E_s(DS)$.
                 u.7 Send $[E_s(m_j) \| MAC_s(M)]$ to the MNO.
                 u.8 Receive $E_s(M_r)$ from the MNO (Step M.3) and decrypt it.

PEERS            P.1 Receive a packet $[E_s(m_i) \| MAC_s(M)] \in E_s(DS)$ (Step u.6).
                 P.2 Apply decision forwarding function (dff).
                 P.3 Send $[E_s(m_i) \| MAC_s(M)] \in E_s(DS)$ to the MNO or forward it to
                     another peer.
                 P.4 Receive $E_s(M_r)$ from the MNO (Step M.3) and delete it.

MNO              M.1 Receive packets (Steps u.7 and P.3).
                 M.2 Forward packets to the SP.
                 M.3 Receive $E_s(M_r)$ from the SP (Step S.4) and forward it to $u$ and
                     PEERS.

SP               S.1 Receive packets from the MNO (Step M.2).
                 S.2 Decrypt the packets and assemble $M$.
                 S.3 Generate and encrypt the response message $E_s(M_r)$.
                 S.4 Send $E_s(M_r)$ to $u$ and PEERS through the MNO.
```

Fig. 4. Anonymous communication protocol

$$dff([E_s(m_i) \| MAC_s(M)]) = \begin{cases} 1 & \text{if } count(MAC_s(M)) = 1 \\ 0 & \text{otherwise.} \end{cases}$$

where *dff*=1 means that the peer under examination has already agreed to send a packet belonging to message M (i.e., with $MAC_s(M)$). If *dff*=1 the peer forwards the received packet m_i to some other peers. Otherwise, if *dff*=0 the peer randomly selects with probability $p_f = \frac{1}{2}$ either to send the packet to the *SP* (white circles in Figure 5(a)) or to forward it to a peer in the communication range (black circles in Figure 5(a)).

Example 1. Figure 5(a) shows an example of the distribution algorithm. The requester u defines $k = 5$ and splits the message M in five parts $\{m_1, \ldots, m_5\}$. Packets are then encrypted with the symmetric key s shared between u and *SP*, and $MAC_s(M)$ is attached to each of them.[5] The requester u selects packet m_3 to be sent directly to the *SP* and forwards the other k-1 packets to peers in the communication range. Specifically, packets m_2 and m_5 are forwarded to peers p_1 and p_3 which send them to the *SP*. Packet m_1 instead takes a forwarded path $p_4 \rightarrow p_7$, assuming p_4 does not accept to send m_1. Finally, packet m_4 takes a forwarded path $p_6 \rightarrow p_7 \rightarrow p_9$ because when the packet is received by p_7, p_7 notices that she has already accepted a packet with the same $MAC_s(M)$ (i.e., m_1) and then automatically forwards m_4 to p_9.

After packets distribution, each selected peer independently sends the packet to the *SP*, through the mobile network operator. The mobile network operator then sees packets that comes from k different users. This scenario results in the following proposition.

Proposition 1. *A user is k-anonymous to the mobile network operator if and only if at least k packets of the same message are sent to the mobile network operator by k different peers (including the requester).*

The mobile network operator forwards the k received packets to the *SP* hiding by default location information. Now, the *SP* can decrypt each packet, reconstruct the original message, and satisfy the user request. A summary of the overall anonymous request process is provided in Figure 5(a).

Anonymous Response. After the conclusion of the anonymous request process, the *SP* retrieves the original message M and starts the service provisioning, which results in the release of an anonymous response to the requester u. The communication involves the mobile network operator to manage peers mobility and route the response to the user u, and must preserve the preference k of the requester.

The anonymous response process works as follow. First of all, as showed in Figure 5(b), the service provider encrypts the response message M_r with the secret key s shared with u. Then the *SP* transmits the encrypted message $E_s(M_r)$ to the k peers involved in the anonymization process. *SP* relies on the cellular network to manage the message delivery and the mobility of the peers. Although all peers receive the message, the

[5] For the sake of clarity, we omit $MAC_s(M)$ in the figure.

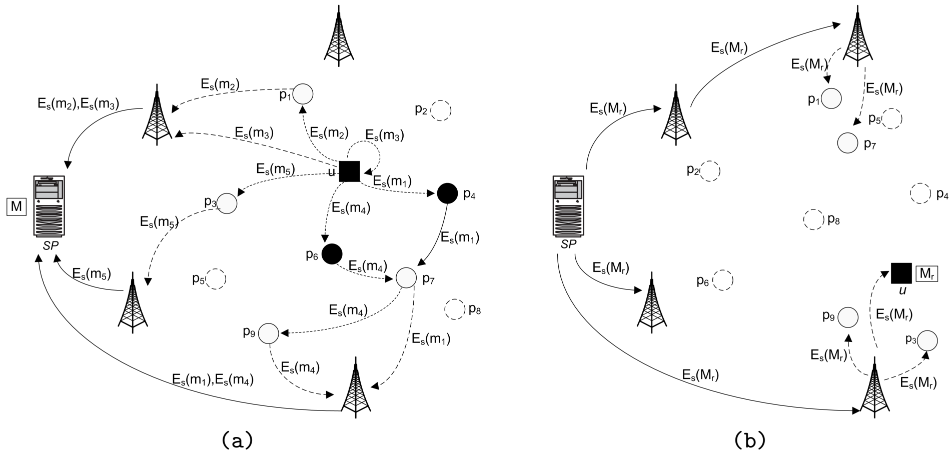


Fig. 5. Example of anonymous request (a) and anonymous response (b)

requester u is the only peer with the secret key s, and thus, she is the only one able to decrypt the message and benefit of the service.[6]

Example 2. Figure 5(b) shows an example of anonymous response. Encrypted message $E_s(M_r)$ is transmitted to all peers used in Example 1, that is, $\{u, p_1, p_3, p_7, p_9\}$. As soon as the message is received by u, it is decrypted. The other peers delete message $E_s(M_r)$, since they are not able to open it.

Recalling the requirements and challenges in Section 3.1, our solution provides both *anonymous communication* and *location hiding*. In terms of anonymous communication, we employ a message splitting and multipath solution that provides k-anonymity against mobile network operators. Considering *location hiding*, the location information of the users is hidden by the cellular network to the service providers. Finally, our solution provides requester accountability, since the requester's identity is released to the service provider.

It is important to note that our solution does not require changes to existing network protocols. All the packets in fact are routed regularly through the hybrid network using TCP and reconstructed at the destination service provider. Only some small changes are requested for specific

[6] To further strengthen our protocol, the service provider could potentially generate $k-1$ decoy messages, other than M_r. This can be performed by adding a *nonce* to the original message M_r before encrypting it with the secret key s. The cellular network sees k different response messages and it is not able to associate the response to the request.

applications on the top of existing layers, as for instance, the message splitting done by the requester u and the packet checks on the mobile ad-hoc network done by the peers.

4 Some Notes on Performance

As a first step, we were interested in quantifying the impact of our approach on the end-to-end communications. Although, this aspect is less significant for database and informational services, it is highly critical for real-time streaming services including video and live operators. Hence, we implemented a prototype of our approach using WiFi-enabled devices and measured the latency overhead when we forward packets to one-hop and two-hop neighbors using WiFi. We describe the testing scenario in Section 4.1 and discuss the performance analysis in Section 4.2.

4.1 Testing Scenario Implementation

We deployed a small-scale testbed using standard IEEE 802.11 communications. We generated two scenarios depicted in Figure 6.

The first scenario (Figure 6(a)) considers baseline measurements in latency of one hop between a *wireless client* and the *target system*. Here, a device is associated directly with a *Wireless Access Point* (WAP); we varied the distance from the client to the WAP. For all practical purposes, the WAP was acting as the one-hop neighbor that forwards the packets to the cellular network.

The second scenario (Figure 6(b)) considers measurements of latency in a two-hop scenario. Here, a device is configured as an *ad-hoc server* on Wireless Adapter #1 (WA1), and with Windows' Internet Connection Sharing (ICS) enabled on Wireless Adapter #2 (WA2), for WA1's traffic. The wireless client is then connected through the ad-hoc server and the WAP to the target system. As in the one-hop scenario, no modification is needed at the WAP. To better simulate a real world scenario, the ad-hoc server has been placed in various locations and distances from the WAP. However, as expected and confirmed by our result, the closer the two systems are to each other, the less latency is observed. Additionally, any implementation in which we have more than one ad-hoc networks should utilize orthogonal channels while broadcasting in the same spectrum, to minimize the interference.

The measurements for both the infrastructure and ad-hoc connections have been taken at approximately the same points. This mitigates variables that might affect WiFi connectivity, such as amount of interference

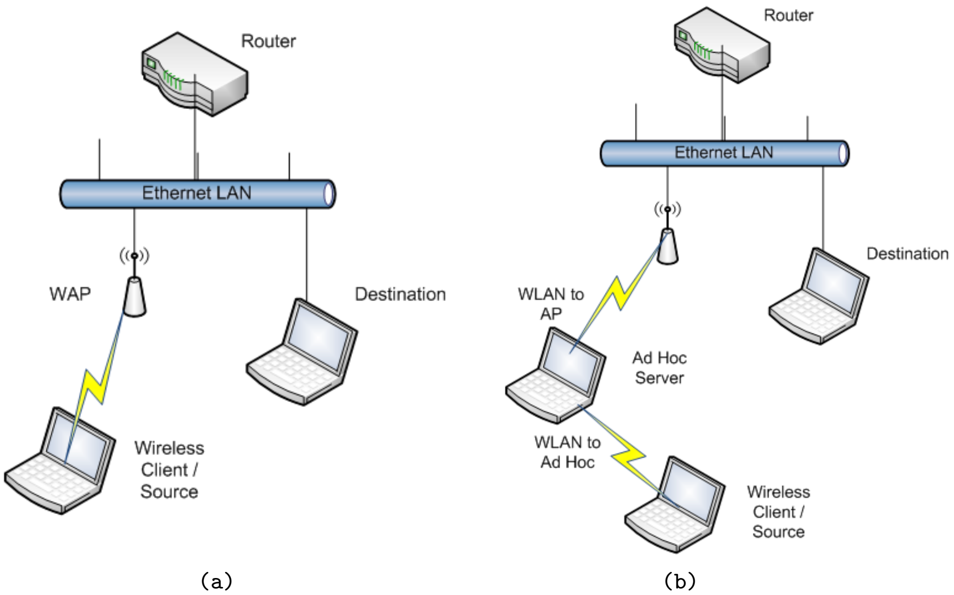


Fig. 6. Network Architecture for One-Hop (a) and Two-Hop (b) Scenarios

from other access points, construction of the building, and obstructions, and seeks to only vary distance/Signal-to-Noise Ratio (SNR).

4.2 Performance Analysis and Discussion

Initially, we measured the Round Trip Time (RTT) of each packet. In addition, we employed NetStumbler [17] to measure the Signal-to-Noise Ratio (SNR), and Wireshark [25] to verify: *i)* packets sent to and received by each device (i.e., the wireless client, ad-hoc server, WAP, and target systems), and *ii)* that the client communication remained anonymous as all packets seemed to originate from the last hop (in this case the WAP).

Table 1 shows our preliminary results. In particular, due to the wireless transmission, we observe a wide variation in latency, mainly due to interference and physical obstacles. Table 1 gives also the average RTT values from which all graphs are evaluated (each value is calculated over more than 25 measurements collected).

The results in Table 1 indicate that there is no significant latency overhead when the SNR is within acceptable bounds. However, the ad-hoc connection becomes much less reliable in weak areas. Figure 7 depicts

Table 1. Maximum, Minimum and Average RTT Values, and Packet Loss Percentages: (a) one-hop WiFi connection, and (b) two-hop WiFi connection

(a)

SNR	RTT (ms)			Packet
	Min	Max	Avg	Loss (%)
14	-	-	-	100
19	3	52	7	0
25	1	28	4	0
28	1	188	63	0
33	1	47	3	0
48	1	33	4	0
55	1	97	23	0
64	1	8	3	0

(b)

SNR	RTT (ms)			Packet
	Min	Max	Avg	Loss (%)
14	-	-	-	100
23	-	-	-	100
31	1	9	3	4
33	1	245	63	0
35	1	13	1	0
47	1	44	5	0
51	1	55	6	3
66	3	104	19	0

scattergraphs of the data sets, with the anomalous peaks representing inconsistencies due to physical obstacles. Figure 7 confirms that peaks in latency due to physical obstacles occur at the same location for all WiFi connections. This is not a measurement inconsistency but rather a verification of the jittery nature of the wireless communications in which physical obstacles affect the transmission even when the distance or the SNR reported by the device remains constant. Moreover, we believe that the RTT measurements are more immediate than the SNR reported by the device which is measured over a period of time. That is why we see this discrepancy of having a good SNR but degraded RTT measurements.

In conclusion, ad-hoc WiFi connections do not seem to suffer much of a performance hit in adding an intermediary node since almost all of our measurements stayed below 5ms of round trip time (or 2.5ms single trip). This allows to safely claim that our system is both deployable and practical even for latency-sensitive applications such as video or voice streaming. However, we must acknowledge that the signal seems unreliable in degraded SNR, so that we might consider using another node with better connectivity. Nodes acting as ad-hoc servers with lower power and bandwidth, such as cellphones instead of laptops, would incur in a performance loss, which may present itself in the form of packet loss and intermittent connectivity, such as was observed in the ad-hoc connection as SNR worsened. While waiting for a ping response, the client node was seen to hang for long periods before announcing an error. This is an issue of QoS because, for example, a page that would attempt to load for some time before displaying an error, or a call that would suspend for some time before finally disconnecting. In using a MultiNet-like technol-

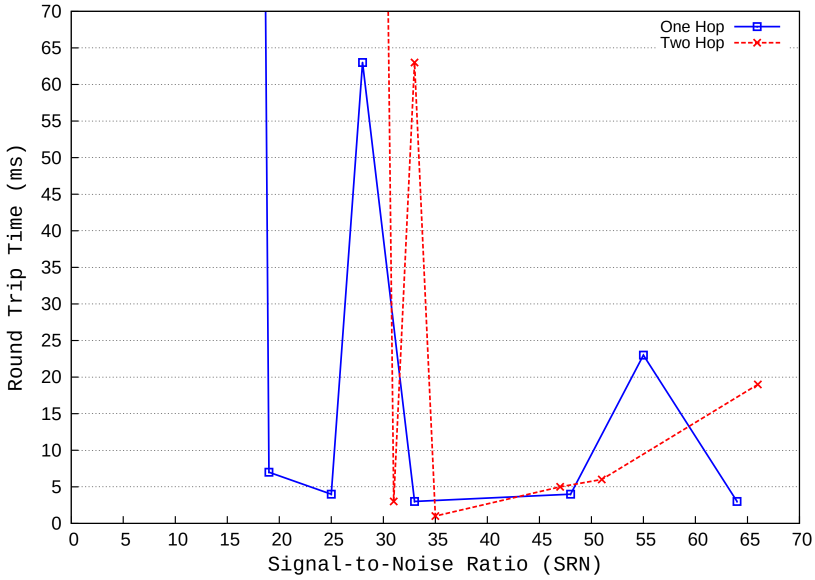


Fig. 7. Comparison Between Graph of SNR vs. RTT (ms) in One-Hop Scenario (in red) and in Two-Hop Scenario (in green). Notice that there is a shift in the graph to the right for two-hops indicating an increase in SNR from the extra hop. Also, there are two symmetric peaks indicating a loss of packets due to physical obstacles.

ogy [23], we could switch between connections, further anonymizing the packet stream. In general, there are a lot of open questions regarding the performance of the entire system, especially under an adversarial model where some of the peers are uncooperative or even malicious.

5 Related Work

While mobile networks and their management have been considered in several works in the area of mobile applications, approaches aimed at protecting the privacy of users have gained great relevance only in the last few years. Recent research in the context of mobile networks approached the privacy problem from different perspectives and have been inspired by works on fully anonymous communications [6, 7, 19].

Anonymous Communications. Chaum introduces the concept of "Mix" to provide source anonymity [6]. A mix collects a number of messages from different sources, shuffles them, and forwards them to the next destination in a random order. This solution makes the tracking of a message difficult for the attackers. In mix-based solution, the path is statically determined by the message sender. Onion routing is a solution that built on the notion of mix network [8]. In onion routing, the connection initiator creates an onion and the path of the connection through the network. Each router (named onion router) in the path knows its successor and can remove a layer of encryption to the onion with its private key. At the end, the data reach the final destination in plaintext. For instance, TOR [8] is an onion routing-based solution that provides route anonymity, by preventing adversaries from following packets from the source to the destination and vice versa. Crowds [19] is an anonymizing solution designed for Web-communications where the routing path and length is dynamically generated. The paths is determined randomly by the machines used in the communications.

An important characteristic shared by the above solutions that makes them not applicable in a mobile scenario like ours is that they use the path generated by the sender for both the request and the response. This assumption cannot be applied in mobile networks where users are moving fast over time and then the path used for the request is likely to be not available both for the response. Also, onion routing solutions are different from our approach because, each onion proxy is required to know the network topology and public certificates of routing nodes to create meaningful routes. Finally, Crowds focuses on protecting the sender's anonymity against the service providers and cannot protect anonymity against a global eavesdropper. Our approach, instead, exploits the hybrid nature of the devices to create a local network which is impervious against global eavesdroppers that operate in the cellular network (e.g., mobile network operators). Since the WiFi network is ad-hoc and of limited range, it is very difficult to have a global eavesdropper that would cover both the WiFi and cellular communications.

Anonymous Mobile Ad-Hoc Routing Protocols. Another line of research has focused on preserving the privacy of wireless traffic by studying and providing privacy-enhanced and anonymous routing protocols. Originally, the proposed mobile ad-hoc routing protocols, such as AODV [18] and DSR [14], were not designed to provide or guarantee privacy and route anonymity but rather they were aimed at increasing network performance, efficiency, security and reliability. As a consequence, in such protocols, there are many ways to compromise privacy; for instance, by abusing the protocol state since both source and destination together with hop-count are stored on each node. Subsequent work focused on routing protocols for mobile ad-hoc networks and attempted to protect anonymity and privacy. They did so by keeping secret the identities of the senders and recipients of messages from intermediate nodes. A number of anonymous routing protocols have been proposed [5, 15, 26, 27, 29, 30]. Among them, MASK [30] proposes an anonymous on demand routing protocol, which provides both MAC-layer and network-layer communications without the need of releasing real identities of the participating nodes. ANODR [15] provides route anonymity, by preventing adversaries from following packets to its source or destination, and location privacy, by preventing the adversary to discover the real identities of local transmitters. Discount-ANODR [27] limits the overhead introduced by ANODR in providing source anonymity and routing privacy. It provides a lightweight protocol based on symmetric key encryption and onion routing. Capkun et al. [20] provide a scheme for secure and privacy-preserving communication in *hybrid* ad-hoc networks. Their scheme provides the users with a means to communicate in a secure environment and preserve their anonymity and location privacy. Although our solution has similar goals and considers privacy issues in hybrid mobile networks, it is not aimed at providing a new routing protocol. Our k-anonymity solution using a multi-path communication paradigm provides privacy of the requester from the neighbors sharing the media, the mobile network operators, and the service providers. Also our solution does not heavily rely on key encryption, dynamic keys or pseudonyms; rather, it exploits the possibility of breaking a single data stream in several different packets, and of using neighbor mobile peers, which act on behalf of the request originator, to distribute these packets.

Location *k*-Anonymity. More recently, another line of research has focused on protecting the location privacy and anonymity of users that interact with Location-Based Services (LBSs) [1, 2]. The main goal of most of the current solutions [16] is to protect users' identities associated with or inferred from location information. In this case, the best possible location measurement can be provided to other entities but users identity must be kept hidden. In particular, these solutions are based on the notion of k-anonymity in data [21], which is aimed at making an individual (i.e., her identity or personal information) not identifiable by releasing a geographical area containing at least k-1 users other than the requester. In this way, the request cannot be associated to fewer than k respondents and the identity of the users is not released to the LBSs. Bettini et al. [3] propose a framework for evaluating the risk of disseminating sensitive location-based information, and introduce a technique aimed at supporting k-anonymity. Gruteser and Grunwald [12] propose a middleware architecture and an adaptive algorithm to adjust location information resolution, in spatial or temporal dimensions, to comply with a specific k-anonymity requirement. Gedik and Liu [10] describe a k-anonymity model and define a message perturbation engine responsible for providing location anonymization of user's requests through identity removal and spatio-temporal obfuscation of location information. Ghinita et al. propose PRIVÈ [11], a decentralized architecture for preserving query anonymization, which is based on the definition of k-anonymous areas obtained exploiting the Hilbert space-filling curve. Hashem and Kulik [13] provide a decentralized approach to location privacy in a wireless ad-hoc network, where each peer is responsible for generating its cloaked area by communicating with others peers, thus providing anonymity. Existing works on location k-anonymity have the following main disadvantages: *i)* they rely on either a centralized middleware for providing anonymity functionalities (centralized approach) or let the burden of the complexity in calculating the k-anonymous area to the users (decentralized approach); *ii)* they assume trusted mobile network operators; *iii)* they do not support accountability. In our approach, we protect the privacy of the users acting in a hybrid network including cellular networks. Here, we assume untrusted mobile network operators, which could track users activities [28], and we provide location k-anonymity at network level rather than at application level.

6 Conclusions and Future Work

We presented a novel privacy-preserving scheme based on network k-anonymity and multi-path that aims at balancing privacy and accountability without assuming any trusted entity between the user and the service provider. Furthermore, we put forward the idea that a reliable privacy solution should protect users against threats stemming from honest but curious mobile network operators. Our vision is then to re-cast privacy for hybrid networks and provide a privacy-assurance mechanism based on network k-anonymity that: *i)* protects users' privacy against honest but curious mobile network operators; *ii)* conceal or obfuscate the users location to service providers, *iii)* enforces user and service accountability. Note that, our solution can be integrated with obfuscation techniques, as the one in [2], to protect the location privacy of the users interacting with LBSs.

Many interesting research directions that warrant further investigation, among which: the enhancement of the decision forwarding algorithms for guaranteeing reliability and efficiency; the consideration of a comprehensive threat model including malicious and uncooperative peers; the complete implementation and extensive testing of our prototype; the consideration of economic incentives for the neighbor peers to participate in our anonymizing network.

Acknowledgments

This work was supported in part by the EU, within the 7th Framework Programme (FP7/2007-2013) under grant agreement no. 216483 “PrimeLife” and by the Italian MIUR, within PRIN 2006, under project no. 2006099978 “Basi di dati crittografate”.

References

1. C.A. Ardagna, M. Cremonini, E. Damiani, S. De Capitani di Vimercati, and P. Samarati. Supporting location-based conditions in access control policies. In *Proc. of the ACM Symposium on Information, Computer and Communications Security (ASIACCS'06)*, Taipei, Taiwan, March 2006.
2. C.A. Ardagna, M. Cremonini, E. Damiani, S. De Capitani di Vimercati, and S. Samarati. Location privacy protection through obfuscation-based techniques. In *Proc. of the 21st Annual IFIP WG 11.3 Working Conference on Data and Applications Security*, Redondo Beach, CA, USA, July 2007.
3. C. Bettini, X.S. Wang, and S. Jajodia. Protecting privacy against location-based personal identification. In *Proc. of the 2nd VLDB Workshop on Secure Data Management*, Trondheim, Norway, 2005.
4. Nikita Borisov, George Danezis, Prateek Mittal, and Parisa Tabriz. Denial of service or denial of security? In *CCS '07: Proceedings of the 14th ACM conference on Computer and communications security*, pages 92–102, New York, NY, USA, 2007. ACM.
5. A. Boukerche, K. El-Khatib, L. Xu, and L. Korba. Sdar: A secure distributed anonymous routing protocol for wireless andmobile ad hoc networks. In *Proc. of the 29th Annual IEEE International Conference on Local Computer Networks (LCN 2004)*, pages 618–624, Tampa, FL, USA, October 2004.
6. D. Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Communications of the ACM*, 24(2):84–88, 1981.
7. D. Chaum. The dining cryptographers problem: Unconditional sender and recipient untraceability. *Journal of Cryptology*, 1(1):65–75, 1988.
8. R. Dingledine, N. Mathewson, and P. Syverson. Tor: The Second-Generation Onion Router. In *Proceedings of the* 13^{th} *USENIX Security Symposium*, pages 303–319, August 2004.

9. T. Fujiwara and T. Watanabe. An ad hoc networking scheme in hybrid networks for emergency communications. *Ad Hoc Networks*, 3(5):607–620, 2005.
10. B. Gedik and L. Liu. Protecting location privacy with personalized k-anonymity: Architecture and algorithms. *IEEE Transactions on Mobile Computing*, 7(1):1–18, January 2008.
11. G. Ghinita, P. Kalnis, and S. Skiadopoulos. Privè: Anonymous location-based queries in distributed mobile systems. In *Proc. of the International World Wide Web Conference (WWW 2007)*, Banff, Canada, May 2007.
12. M. Gruteser and D. Grunwald. Anonymous usage of location-based services through spatial and temporal cloaking. In *Proc. of the 1st International Conference on Mobile Systems, Applications, and Services (MobiSys 2003)*, San Francisco, CA, USA, May 2003.
13. T. Hashem and L. Kulik. Safeguarding location privacy in wireless ad-hoc networks. In *Proc. of the 9th International Conference on Ubiquitous Computing (UbiComp 2007)*, Innsbruck, Austria, September 2007.
14. D. B. Johnson and D. A. Maltz. *Dynamic Source Routing in Ad Hoc Wireless Networks*, volume 353. Kluwer Academic Publishers, 1996.
15. J. Kong and X. Hong. ANODR: Anonymous on demand routing with untraceable routes for mobile ad-hoc networks. In *Proc. of the 4th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC 2003)*, pages 291–302, Annapolis, MD, USA, June 2003.
16. S. Mascetti and C. Bettini. A comparison of spatial generalization algorithms for lbs privacy preservation. In *Proc. of the 1st International Workshop on Privacy-Aware Location-based Mobile Services (PALMS 2007)*. IEEE Computer Society, 2007.
17. *NetStumbler.com*. `http://www.netstumbler.com/`.
18. C.E. Perkins and E.M. Royer. Ad-hoc on demand distance vector routing. In *Proc. of the 2nd IEEE Workshop on Mobile Computing Systems and Applications (WMCSA99)*, New Orleans, LA, USA, February 1999.
19. M.K. Reiter and A.D. Rubin. Crowds: Anonymity for web transactions. *ACM Transactions on Information and System Security*, 1(1):66–92, 1998.
20. J.-P. Hubaux S. Capkun and M. Jakobsson. *Secure and Privacy-Preserving Communication in Hybrid Ad Hoc Networks*, January 2004.
21. P. Samarati. Protecting respondents' identities in microdata release. *IEEE Transactions on Knowledge and Data Engineering*, 13(6):1010–1027, 2001.
22. *Sphinx - A Hybrid Network Model for Next Generation Wireless Systems*. `http://www.ece.gatech.edu/research/GNAN/work/sphinx/sphinx.html`.
23. Angelos Stavrou and Angelos D. Keromytis. Countering dos attacks with stateless multipath overlays. In *CCS '05: Proceedings of the 12th ACM conference on Computer and communications security*, pages 249–259, New York, NY, USA, 2005. ACM.
24. L. von Ahn, A. Bortz, and N.J. Hopper. k-anonymous message transmission. In *Proc. of the 10th ACM Conference on Computer and Communication Security (CCS 2003)*, pages 122–130, Washingtion, DC, USA, October 2003.
25. *Wireshark*. `http://www.wireshark.org/`.
26. X. Wu and B. Bhargava. Ao2p: Ad hoc on-demand position-based private routing protocol. *IEEE Transaction on Mobile Computing*, 4(4), July/August 2005.
27. L. Yang, M. Jakobsson, and S. Wetzel. Discount anonymous on demand routing for mobile ad hoc networks. In *Proc. of the Second International Conference on Security and Privacy in Communication Networks (SECURECOMM 2006)*, Baltimore, MD, USA, August-September 2006.

28. C. Zander. *'CIA CELL TOWER' Monitors Local Internet Users' Wireless Transmissions*, September 2007. `http://www.send2press.com/newswire/2007-09-0911-003.shtml`.
29. Y. Zhang, W. Liu, and W. Lou. Anonymous communication in mobile ad hoc networks. In *Proc. of the 24th Annual Joint Conference of the IEEE Communication Society (INFOCOM 2005)*, Miami, FL, USA, March 2005.
30. Y. Zhang, W. Liu, W. Lou, and Y. Fang. Mask: Anonymous on-demand routing in mobile ad hoc networks. *IEEE Transaction on Wireless Communications*, 5(9), September 2006.

User Privacy in Transport Systems Based on RFID E-Tickets

Ahmad-Reza Sadeghi[1], Ivan Visconti[2], and Christian Wachsmann[1]

[1] Ruhr-University Bochum
Horst-Görtz Institute for IT-Security (HGI), Germany
{ahmad.sadeghi,christian.wachsmann}@trust.rub.de

[2] Dipartimento di Informatica ed Applicazioni
University of Salerno, Italy
visconti@dia.unisa.it

Abstract. Recently, operators of public transportation in many countries started to roll out electronic tickets (e-tickets). E-tickets offer several advantages to transit enterprises as well as to their customers, e.g., they aggravate forgeries by cryptographic means whereas customers benefit from fast and convenient verification of tickets or replacement of lost ones.

Existing (proprietary) e-ticket systems deployed in practice are mainly based on RFID technologies where RFID tags prove authorization by releasing spatio-temporal data that discloses customer-related data, in particular their location. Moreover, available literature on privacy-preserving RFID-based protocols lack practicability for real world scenarios.

In this paper, we discuss appropriate security and privacy requirements for e-tickets and point out the shortcomings of existing proposals. We then propose solutions for practical privacy-preserving e-tickets based on known cryptographic techniques and RFID technology.

Key words: Location Privacy, E-Tickets, RFID

1 Introduction

Electronic tickets (e-tickets) gain increasing popularity among operators of public transit networks. However, besides offering many advantages, e-tickets also introduce several risks, in particular concerning privacy of their users.

Benefits of e-tickets. Transit enterprises benefit from e-tickets in various ways: First, e-tickets help to decrease maintenance costs. Second, the number of fare dodgers is expected to decrease if tickets can be verified efficiently. Moreover, cryptographic means help to aggravate the problem of ticket forgery.

From the user perspective, e-tickets allow for faster and more convenient verification. Moreover, an e-ticket system can automatically select the lowest fare, which saves the customer's time and money. Finally, revocation of e-tickets enables transit enterprises to replace lost tickets, which is not possible for conventional paper-based ticket systems.

Threats. Besides their advantages, e-tickets also introduce several risks, in particular regarding the privacy of users. Since authentication of transit tickets typically involves spatio-temporal data, users are at risk to loose their privacy if this information is leaked to unauthorized parties. This means that e-tickets should ensure that no information on users (*confidentiality*) or their movements (*location privacy*) should be revealed to entities that are not trusted by the users. There are existing implementations of e-tickets that allow the creation of movement profiles and, in some cases, even disclose personal information of users (cf. Section 3). Moreover, since e-tickets contain digital data, they may be easily copied (*cloning*). Additionally, the corresponding protocols to issue or verify e-tickets may be subject to different attacks (e.g., man-in-the middle or replay).

Current situation. Currently, there is a vast amount of existing proprietary solutions for e-tickets. Since the corresponding specifications are usually not publicly accessible, there is no publicly known solution in practice that explicitly considers the privacy of users. We stress that user privacy preservation has not been claimed among the features of such systems.

The preferred technology to implement electronic transit tickets is Radio Frequency IDentification (RFID), which enables fully automated wireless identification of objects. A typical RFID system consists of *transponders* and *transceivers*. The main component of a RFID system is the transponder, which consists of an integrated circuit that is connected to an antenna. Typically, transponders are integrated into plastic cards or stickers that can be attached to the object to be identified and thus are often called *tags*. Since transceivers are mainly used to read data from tags, they are called *readers*. RFID tags can be used to realize e-tickets that are issued and verified by readers. Thus, in the rest of this paper "e-ticket" refers to tickets based on RFID.

Related work. There is a large body of literature on different approaches to realize privacy-preserving mechanisms for RFID (e.g., [17,16,2,31,14,20,22,9,18,25]). However, as pointed out in Section 3, most of these solutions are not applicable to e-tickets since each of them lacks some important security and functional requirements, as usability, security and privacy.

In [15], the authors motivate research for privacy in the context of e-tickets and provide a rough description of how anonymous credential [6] and e-cash [5] systems may be used to implement an anonymous payment system for public transit. However, they assume that devices realizing tickets can perform computationally demanding protocols (i.e., use public-key cryptography and intensive interaction), which is not a reasonable assumption for currently available cheap RF tokens. Since RFID tags are devices with very limited capabilities, one has to provide an acceptable level of privacy still preserving usability.

Summing up, an e-ticket system is an authentication scheme that involves spatio-temporal information and the design and secure implementation of a privacy-preserving and usable system based on RFID, is currently an interesting open problem.

Our contribution. In this paper we study the levels of privacy that could be achieved in an e-ticket powered system. We point out the weaknesses of known solutions and explore how known cryptographic tools can be applied to realize anonymization of e-tickets with currently available RFID technology, while having the goal to obtain a usable system that ensures no information disclosure on the user or his location to entities that are not trusted by the user.

Structure of the paper. In Section 2, we demonstrate the problems related to e-tickets by introducing the setting of electronic transit tickets, and define appropriate security requirements. In Section 3, we analyze several proposals from literature on how to realize anonymity for RF-tokens with limited capabilities and discuss their applicability to e-tickets. Section 4 describes how recent cryptographic tools can be applied in order to achieve the desired requirements. Finally, we conclude with Section 5 by describing some open problems and motivating further research.

2 Scenario of Electronic Transit Tickets

To introduce the problems related to e-tickets for public transportation, we first give a short overview of the general application scenario and point out potential weaknesses.

2.1 General Application Scenario

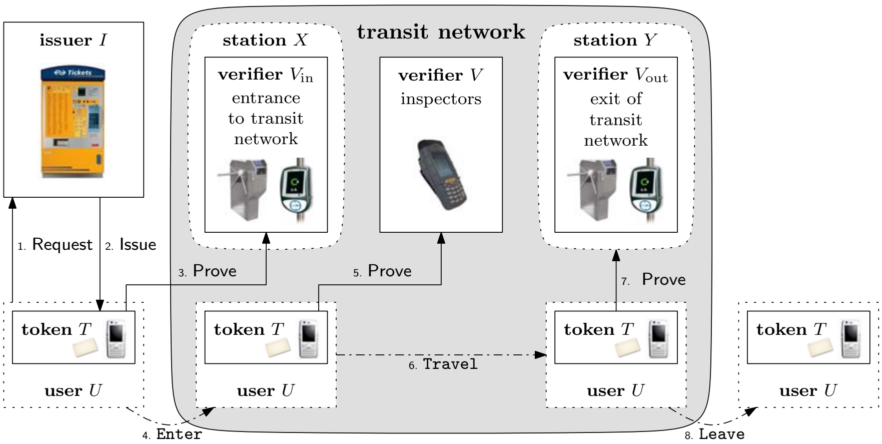


Fig. 1. General scenario for e-tickets.

An e-ticket system as shown in Fig. 1 is a *token-based authentication scheme* whereas tickets are represented as tokens (e.g., RFID tags). It consists of at least one token issuing entity (*issuer*), a set of *users*, *tokens*, and *verifiers* who verify whether tokens are valid.

Typically, a user U must buy a token from token issuer I. Therefore, U selects his desired ticket and pays it. Issuer I then checks whether U is eligible to obtain a token (e.g., whether U paid for the ticket), and, if applicable, issues a token T and passes it to U. From now on, U is able to use token T to prove that he is authorized to use the transit network. This means that every user who is in possession of a token that has been issued by a genuine issuer is considered to be an *authorized user*.

Now assume that, as shown in Fig. 1, user U wants to travel from a place X to some location Y. Before U is allowed to enter the transit system at X, he must first prove to a verifier V_{in} at the entrance of the transit network that he is authorized to access it. If V_{in} can successfully verify the user's token, U is allowed to enter. Otherwise access will be denied. During his trip, U may encounter arbitrary inspections where he must prove that he is authorized to use the transit network. Thus, a verifier V may check the user's token T. If verification of T is successful, U is allowed to continue his trip. Otherwise, U must leave the transit network and may be punished for using it without authorization. After arriving at Y, the user's token T can be checked for a last time. Again, if T cannot be verified successfully, U may be punished.

Note that authentication is typically bound to some limitations. For instance, this may be some geographical or timely usage restrictions. Additionally, a token may be bound to the identity of its *owner* (i.e., the entity that bought the ticket).

2.2 Potential Attacks

Obviously, the main goal of a ticket system is to prevent ineligible users from using the transit system. Thus, the most prominent attack is to violate this goal. However, there are some other, subtle attacks which we are going to consider in the following.

Impersonation. The most obvious attack against e-ticket systems is motivated by unauthorized entities. The adversary must obtain or simulate a token that is accepted by an honest verifier. To achieve this, the adversary may perform various attacks including man-in-the-middle or replay attacks against the underlying authentication protocols, or he may attempt to create forged tokens, or to copy tokens of honest users.

Tracing. A more subtle attack aims at obtaining information on users and their movements within the transit network. For instance, the transit enterprise may be interested in information on the behavior of its customers. When using conventional authentication protocols, a token can be easily identified during verification. This enables verifiers to trace tokens within the transit network. Moreover, if a user uses an identifying payment method (e.g., a credit card) to buy a token, the issuer can link the token to the identity of its owner. Since the issuer and the verifiers are typically under the control of the same entity (e.g., the transit enterprise), this results in a complete loss of the user's privacy. However, in this case user information is managed by the transit enterprise that is a known entity. Thus it can be subject by law to commit on the honest use of the collected user data and can be monitored by means of inspections (similar observations hold for credit card companies).

The concrete threat instead, comes from unknown adversaries. Tokens typically are wireless devices and thus all their communication can be eavesdropped or manipulated by an adversary. Moreover the adversary may unnoticeably interact with tokens. As a consequence, the user's token may also be traced by entities different from the verifiers or the token issuer.

In summary, a primary goal is that the e-ticket system prevents disclosure of information on users or their movements to entities not trusted by the users.

Denial-of-service attacks. Another type of adversary may want to harm (e.g., to blackmail) the transit enterprise by preventing honest users from accessing the transit network. As already mentioned, tokens are wireless devices that can be attacked unnoticeably. This means that an adversary may try to exploit deficiencies of the protocols such that a ticket is no longer accepted by an honest verifier.

Depending on the underlying business model, protocols for e-tickets must be carefully crafted to prevent some or all of these attacks. In Section 2.3, we introduce different reasonable trust and adversary models and set up a complete list of requirements for e-ticket systems in Section 2.4.

2.3 Trust and Adversary Model

In an ideal setting, no entity must be trusted. However, in practice, the transit enterprise must at least trust issuer I to only create tokens for eligible users. Moreover, the transit enterprise must trust each verifier V to only accept tokens that have been issued by issuer I. These are reasonable assumptions since in practice, the token issuing entity and the verifiers are typically physically controlled by the transit enterprise.

Ideally, users should be anonymous to every entity, including issuer I and all verifiers V. However, due to technical restrains this is not always feasible in practice. Thus, a reasonable trust model for a practical solution is that users must at least trust issuer I and, dependent on the implementation, also all verifiers V. However, a trust model which only requires issuer I to be trusted is preferable.

To summarize, issuer I must trust all verifiers V. Moreover, all verifiers V must trust token issuer I. For users, there are three possible trust models:

TM 1: User U must trust token issuer I and all verifiers V.
TM 2: User U must only trust token issuer I.
TM 3: User U needs not to trust anyone.

TM 1 means that the e-ticket system must preserve privacy to all entities outside the system. This is the trust model primarily used for the solution presented in Section 4. Considering TM 2, the e-ticket system must additionally protect the user's privacy to the verifiers. The solution presented in Section 4 can achieve this by assuming each verifier V to be connected to a remote server or to be equipped with a security module that is controlled by issuer I. However, these hardware assumptions may be difficult to achieve in practice. To realize TM 3, the e-ticket scheme must provide full anonymity. As discussed in Section 1, this seems to be possible only with high computational and communication resources, which is inappropriate for low-cost RFID devices.

It is also assumed that all communication that takes place during the process of creating a ticket cannot be eavesdropped or manipulated by an adversary. This is reasonable in practice since a user U may either use out-of-band communication or a secure channel to communicate to issuer I. However, following the traditional adversarial models, an adversary can eavesdrop all communication of a token T. Moreover, an adversary may perform active attacks on the corresponding protocols, which means that he can interact with all parties on the protocol level. Additionally, an adversary can corrupt tokens and verifiers (though this can only happen for a limited number of tokens and verifiers). The adversary is not allowed to corrupt the token issuer.

2.4 Requirement Analysis

Authentication. As mentioned in Section 2.1, the most important security goal for transit enterprises is authentication. Thus no unauthorized user (i.e., who is not in possession of a valid token) should be able to convince an honest verifier that he is authorized to access the transit system.

Another major requirement for any token-based authentication scheme is the resilience to remote tampering with tokens, which would allow denial-of-service attacks.

We summarize the security goals concerning authentication as follows:

Authentication: Only valid tokens are accepted by honest verifiers.
Unforgeability: Emulation and copying of valid tokens should be infeasible.
Availability: Unauthorized altering of token data must be infeasible.

Privacy. Since e-tickets enable efficient detection and identification of a huge number of tickets, a detailed dossier about user profiles (e.g., personal data or movements) can be created. The problem aggravates if tickets can be associated with the identity of their corresponding users since this results in a complete loss of user privacy.

Thus, the security objectives concerning privacy are:

Confidentiality: Unauthorized access to user-related data should be infeasible.
Anonymity: Unauthorized identification of tokens should be infeasible.
Location Privacy: Unauthorized tracing of tokens should be infeasible.

A stronger notion of location privacy considers traceability of tokens in case the internal state (i.e., the secrets) of a token has been disclosed. To distinguish traceability in past or future protocol runs, [18] consider the notion of *forward* and *backward traceability.*

Backward traceability: Accessing the current state of a token should not allow to trace the token in previous protocol runs.
Forward traceability: Accessing the state of a token should not allow to trace the token in future protocol runs.

In addition to these security and privacy requirements it is important to consider functional requirements for a practical solution.

Functional requirements. The costs per e-ticket should be minimal. Therefore, in case each ticket is implemented as a physical token (e.g., as RFID tag), the computational and storage requirements to the token should be as low as possible.

Additionally, verification of tickets must be fast. For instance, it should be possible to verify an e-ticket while a user is walking by, or shortly holding his ticket near a verifying device (e.g., while entering a bus). Therefore, protocols for e-tickets must be designed carefully to minimize the amount of computation and communication that must be performed. Moreover, an e-ticket system must be able to handle a huge amount of tokens.

Therefore, the functional requirements to e-tickets are:

Efficiency: Verification of tokens must be fast.
Scalability: The system should be able to handle a large amount of tokens.

Depending on the underlying business case and the technological restraints a practical realization may not fulfill all of these requirements.

3 Analysis of Existing Solutions

Most e-ticket systems are proprietary solutions whose specifications are not publicly available. This section exemplary shows the most common approach of implementing authentication of e-tickets in practice by the Calypso e-ticket system [1,26], of which at least some information is public. Moreover, to the best of our knowledge, there is no solution for e-tickets in practice that explicitly considers privacy of users.

Calypso e-ticket standard. Calypso is an e-ticket standard based on RFID tokens that is widely used in Europe and North and South America [1]. The roles in the Calypso system correspond to the model presented in Fig. 2. However, Calypso does not consider privacy of users and thus does not fulfill any of the privacy requirements of Section 2.4 w.r.t. any of the trust models presented in Section 2.3. In fact, all transactions involving a Calypso e-ticket provide no confidentiality at all [26]. Moreover, Calypso tokens store personal data of their owner ("holder information") that can be queried by every verifier. Thus the Calypso e-ticket system leaks user-related information and allows the creation of movement profiles by everyone who is in possession of a standard RFID reader. However, all messages of a Calypso token are authenticated by a symmetric-key-based authentication mechanism. Thus, Calypso seems to fulfill all of the authentication requirements of Section 2.4.

Calypso implements a common approach to authenticate a low-cost RFID token based on a simple challenge-response protocol. Each token has a symmetric authentication key K_T that can be computed as a function of the serial number S_T of the token and a global master secret. All verifiers are equipped with a tamper-resistant security module (secure application module, SAM) that knows and protects this master secret and can be used as a black-box to compute K_T from S_T. To authenticate a token, a verifier sends a random challenge N_V to the token,which then computes $H_T \leftarrow f(K_T, N_V)$ where f is some one-way function. Finally, the token returns (S_T, H_T) to the verifier who uses its SAM to drive K_T and then verifies H_T. If verification is successful, the token has been authenticated. Obviously, this approach cannot provide privacy since all transactions of a token can be linked by its serial number S_T that is transmitted in clear in every protocol run. All subsequent transactions to update or to read data from a Calypso token are authenticated this way but are not encrypted.

Other e-ticket systems. There are many other proprietary solutions for e-tickets in practice. Most of them are based on widely used RFID transponders. Prominent examples are FeliCa [11] and MiFare [24].

FeliCa [11] is provided by Sony and is a contactless smartcard that is used mainly in the Asia-Pacific area for different purposes including e-tickets for public transportation.

MiFare is a family of contactless smartcards produced by Philips/NXP Semiconductors. These transponders are widely used for different purposes including e-tickets for public transportation. There were several publications on attacks against MiFare Classic transponders [21,23], that use a proprietary encryption algorithm that has been completely broken [7]. However, other MiFare products are claimed not to be affected.

The attacks on MiFare Classic transponders demonstrate a major problem of proprietary security solutions: Manufacturers of low-cost hardware try to find a compromise between speed and security of their products. Thus, they often implement proprietary lightweight crypto algorithms whose specifications are not public, and thus are typically not sufficiently evaluated. As the attack against MiFare Classic shows, these algorithms can often be reverse-engineered, which allows cryptanalysis or efficient key search by running the algorithms on more powerful hardware. In case of MiFare Classic, both ways enabled to break the security goals of these tags at a point in time where they were already widely used in practice.

3.1 Protocols for Anonymous Authentication

In an ideal e-ticket system, verifiers should learn nothing from the verification except that a token is genuine and valid. It is possible to realize this by using privacy-preserving techniques like anonymous credential systems [15]. An anonymous credential system is a cryptographic tool that enables zero-knowledge proofs of knowledge of certified data [19]. However, using anonymous credentials implies high computational (public-key cryptography) and typically also high communication (many rounds of interaction) requirements to all devices involved. Apparently, this is a contradiction to the functional requirements described in Section 2.4. Thus, these techniques are not applicable unless the e-ticket system can fall back upon appropriate mobile computing devices that are already possessed by the users. However, using mobile computing devices, like mobile phones, has several disadvantages. For instance, in case a user's phone runs out of power (which probably happens very often) he will no longer be able to prove authorization. Moreover, these devices can also be compromised by Trojans, which brings up new challenges. Furthermore, many users do not yet own a NFC[3] compatible mobile phone that has sufficient computing power to run computationally demanding protocols like anonymous credential systems or e-cash as proposed in [15].

3.2 Privacy-Preserving Protocols for RFID

There is a large body of literature on different approaches to implement privacy-preserving mechanisms for low-cost RFID transponders. For instance, [16] gives a comprehensive overview of different approaches. The author classifies RFID transponders as basic tags and symmetric-key tags. *Basic tags* refers to tokens that have no computational and no cryptographic capabilities. *Symmetric-key tags* means tags that are capable of performing at least some symmetric cryptographic functions (e.g., random number generation, hashing, or encryption). Using the classification of [16], we discuss the applicability of different proposed solutions to e-tickets.

Basic tags. As basic tags cannot perform any cryptographic operations they disqualify for authentication purposes. Tags that only provide wireless readable memory can only forward the data stored in their memory and thus are subject to replay and cloning attacks. This means that all data stored on such a tag can be read and be used to create identical copies or to simulate the original tag to an honest reader. Another problem related to cloning is *swapping.* This means that an adversary can copy the data stored on tag A to another tag B and vice versa and thus change the identities of these tags. Therefore, basic tags cannot fulfill the requirement of *unforgeability.*

[3] Near Field Communication (NFC) [10] is a RFID standard for contactless smartcards that is also supported by some currently available mobile phones.

Moreover, many solutions to enhance privacy of basic tags require tags to provide *many-writable* memory (e.g., [17,13,2]). The basic idea of these schemes is to frequently update the information stored on the tags such that an adversary cannot link them. However, due to the lack of secure access control mechanisms it is impossible to prevent unauthorized writes to such tags. A simple denial-of-service attack is to write some garbage data to a tag. Thus, an honest verifier will no longer accept the tag until it is reinitialized with correct data. This violates the *availability* requirement.

Therefore, tags that provide no cryptographic functionality cannot be used in applications that require reliable authentication. Thus, it is inevitable to use tags that are capable of performing at least some cryptographic functions if authentication is of concern.

Symmetric-key tags. A general problem of implementing privacy-preserving authentication based on symmetric keys is how to inform the other party which key must be used. Apparently, a tag cannot disclose its identity before the reader has been authenticated since this would violate its *location privacy*. Therefore, the reader does not know which authentication key it should use, and thus cannot authenticate to the tag. The basic idea to circumvent this problem has been introduced by [31] as *Randomized Access Control*:

Let $f_K(m)$ be a keyed one-way function on message m using key K. To authenticate to a reader, a tag first computes $h_T \leftarrow f_{K_T}(R)$ where K_T is a tag-specific key and R is a random value chosen by the tag. On receipt of (h_T, R), the reader forwards this tuple to a trusted server that computes $h_i \leftarrow f_{K_i}(R)$ for all keys $K_i \in \mathcal{K}$ where $\mathcal{K}$ denotes the set of the keys of all authorized tags. The server accepts if it finds a $K_i \in \mathcal{K}$ such that $h_i = h_T$. Finally, the server sends its decision whether to accept or reject the tag to the reader. Since R is randomly chosen each time the tag is queried, it always emits a different tuple (h_T, R) which cannot be linked to the tuples sent in previous protocol runs. Moreover, the reader does not learn the identity (i.e., the key K_T) of the tag since it only receives the response from the server. An obvious drawback of this solution is that the computational costs for the server to verify a tag are linear in the number of authorized tags. Therefore, this basic approach does not fulfill the *efficiency* and *scalability* requirement. Another disadvantage of this solution is that readers must have an online connection to the server, which, depending on the use case, may not be practical. Moreover, the tag must trust the server to respect its privacy since the server can identify the tag when it found the right key. Furthermore, this solution provides no security against replay-attacks and thus violates the *unforgeability* requirement. There is many subsequent work (including [20,9,18,25]) that follows and optimizes this approach by introducing new setup assumptions or by lowering the security or privacy requirements.

Other approaches rely on updating the identity of a tag each time it has been authenticated [14,27]. These approaches allow authentication of a tag in constant time. However, they require the verifiers to have permanent access to a trusted database that verifies tags for them and manages all updates of the tag identities. As discussed above, this may be inappropriate for e-ticket systems.

Section 4 provides a simple solution that allows anonymous authentication of tags with constant computational costs for the readers without the need for a permanent online connection.

4 Solution for Practical Privacy-Preserving E-Tickets

RFID tags that are capable of performing public-key operations disqualify for practicable implementations of e-tickets because of their relatively high price and low performance. Thus, RFID tokens that are limited to symmetric-key cryptography (i.e., random number generation and hashing) are the most practical choice for e-tickets. However, as discussed in Section 3.2 the use of symmetric-key cryptography seems to have the drawback that at least the token issuer must be trusted not to disclose personal information or movement profiles of users. Thus, our solution is based on the trust and adversary model for e-tickets that we discuss in the following.

Trust and Adversary Model. Following Section 2.3, for e-tickets based on RFID tokens that are limited to symmetric cryptography, either trust model TM 1 or trust model TM 2 must be chosen. This means that a user U must at least trust token issuer I. Whether user U must additionally trust all verifiers V depends on the corresponding setup assumptions. This means that, if verifiers are considered to be untrusted, all operations that disclose user-related information must be dropped from the verifiers. For instance, these computations may be carried out on a local tamper-resistant[4] security module as it is done by many implementations in practice (cf. Section 3). Another simple approach used by various anonymous symmetric-key-based authentication protocols, is to employ a remote trusted server (cf. Section 3.2).

Model for Anonymous E-Ticket Systems. As discussed in Section 2.2, to provide privacy of users it is necessary to prevent tracing of tokens. This means that all entities that are not trusted by the user of a token should not be able to decide whether the user's token has been used in a protocol run (*unlinkability*).

Therefore, it is necessary to employ some mechanism that hides the identity of a token each time it is queried. This can either be some special hardware (e.g., as proposed by [2]) or a cryptographic primitive that inherently provides anonymity of users (e.g., anonymous credentials as proposed in [15]). In the following, we refer to this mechanism as *anonymizer*.

Analogous to Section 2.1, an anonymous e-ticket system consists of at least one token issuer, a set of users, tokens, verifiers, and anonymizers. The token issuer creates tokens for users. These tokens can be used by users to prove to verifiers that they are authorized to use the transit system. Additionally,

[4] Tamper-resistance means that the device will delete all its secrets when it detects any kind of physical tampering.

anonymizers ensure anonymity of tokens. We say that tokens are *anonymized.* Fig. 2 illustrates the model for anonymous e-ticket systems.

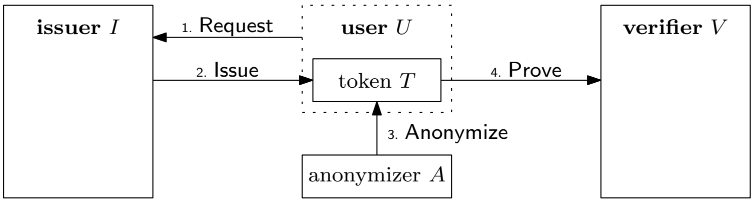


Fig. 2. Model for anonymous e-ticket systems.

Description of the Solution. In the following, we focus on solutions for privacy-preserving authentication based on RFID tokens that are at most capable of performing symmetric cryptography.

The players are as shown in Fig. 2. The anonymizer is either a dedicated hardware device or a software running on a mobile computing device (e.g., the mobile phone) of the user. Note that, a separate anonymizer device may suffer from the same problems as discussed in Section 3.1. However, in case a user's anonymizer runs out of power, the user will indeed loose some privacy until his anonymizer is operable again but he can still prove authorization using his RFID token. Moreover, since anonymizers can also be available in public places and their capabilities can be embedded in the verifiers (when this does not significantly affect the performance of the system), the user's privacy is not completely lost.

Since our solution relies on symmetric-key-based authentication, the token must store an authentication secret. To achieve the security requirement of *unforgeability*, it should be impossible to determine this secret by attacking the protocols involving the token, as well as by physically attacking the token. One solution to counterfeit physical attacks is to employ physical protection mechanisms that aggravate reading out the memory of the tag. However, this would increase the price of tag such that it would be improvident to use them. Another solution to prevent cloning can be implemented by means of a recent physical cryptographic primitive: Physically Unclonable Functions (PUFs) [28,29].

4.1 Building Blocks

Secure key storage with PUFs. A *Physically Unclonable Function* (PUF) is an inherently unclonable function embedded into a physical object [29]. The unclonability of the PUF comes from random and uncontrollable manufacturing processes during creation of the corresponding object. A PUF maps challenges to responses. A *challenge* is a stimulus that, when applied to the PUF makes it to return a *response* that is specific for the PUF w.r.t. to the stimulus. Since the response of a PUF relies on physical properties of the corresponding physical object, which is subject to noise (e.g., temperature, pressure, etc.), the PUF will always return slightly different responses to the same stimulus.

A PUF can be embedded into a microchip, e.g., by exploiting statistical variations of delays of gates and wires within the chip. These deviations are unique for every sample from a set of chips that implement the same circuit. Therefore, in [28], the authors propose to use a PUF as secure key storage.

The adversary model for PUFs is that an attacker is assumed to know how the PUF is challenged and how responses are measured. Moreover, the attacker is allowed to know the exact challenges for deriving the secret stored in the PUF. The requirements to the chip that incorporates a PUF to securely store a secret are as follows [29]:

1. The PUF must be inseparably bound to the chip such that any attempt to separate them results in significant damage to the PUF and the chip.
2. Any measurements to the chip must not reveal detailed information on the structure of its PUF.
3. The PUF, the sensors for measuring responses, the processing unit, and the volatile memory of the chip must be opaque.
4. Even if details on the structure of the PUF are known, it must be infeasible to create a physical copy or to set up a mathematical model of the PUF that allows to predict challenge-response pairs with non-negligible probability (*unclonability*).
5. Tampering with the chip or the PUF must significantly change the challenge-response behavior of the PUF (*tamper-evidence*).
6. The chip must contain tamper-proof read-only memory that stores public data (e.g., algorithms) whose integrity is important.

The first requirement prevents an adversary from accessing the output of (and thus, the secret stored in) the PUF. The second prevents an adversary from collecting data that may help to create a clone or to set up a mathematical model of the PUF that can be used to obtain the secret. The third is to prevent any kind of attacks (e.g., side-channel attacks) that try to disclose the internal state of the PUF, the processing unit, or the volatile memory of the chip that may temporally contain parts of the secret. The fourth is to prevent cloning of the PUF. The fifth prevents invasive inspections, which means that any attempt to physically access or manipulate the chip or the PUF must destroy both of them. The last requirement prevents an attacker from injecting malicious code that may force the chip to disclose its secret.

Storing a secret in a PUF. To use a PUF as a secure key storage, a key K is generated and stored as follows: A trusted party (e.g., issuer I) first generates key $K \in_{\mathcal{R}} \{0,1\}^l$ using an appropriate security parameter l. Then, it chooses a random challenge z to challenge the PUF. On receipt of response r, the trusted party computes some helper data w such that key K can later be recovered by evaluating $\mathsf{PUF}(z, w)$ and stores (z, w, K) in a database. The tuple (z, w) is stored in the (unprotected) memory of the chip.

Helper data w has two different purposes [29]: First it should help to remove the effects of noise on measurements of the responses of the PUF, and second, since the responses of a PUF are typically not uniformly distributed, w should guarantee that secret K is uniform.

Reconstructing a secret from a PUF. To reconstruct secret K, the chip reads (z, w) from its memory, challenges its PUF with (z, w) and obtains K.

Efficient implementation. According to [28], a PUF can be integrated into a chip with less than 1000 extra gates. Moreover, [8] presents an implementation of a PUF for RFIDs.

Symmetric-key-based authentication. In order to authenticate tokens, standard authentication mechanisms based on symmetric-key cryptography that are secure against impersonation under passive (imp-pa) and active (imp-aa) attacks [4, p. 10] can be used. Since low-cost RFID tags are not capable of running multiple sessions, concurrent attacks (imp-ca) must not be considered. To provide *confidentiality* and *location privacy*, the authentication scheme must not disclose user-related information (e.g., user data or movement profiles).

As described in Section 3.2, the major problem of realizing anonymous authentication based on shared secrets is how to inform the other party about which secret should be used without revealing the own identity. This problem can be solved by employing rerandomizable public-key encryption [13,2].

Rerandomizable encryption. A *rerandomizable encryption scheme* means an encryption scheme for which there is a probabilistic function $\mathsf{Rand}(\cdot)$ that maps ciphertexts c to ciphertexts $c' \neq c$ such that the corresponding plaintext stays the same. The rerandomizable encryption scheme must be semantically secure [12] and should provide key privacy [3]. Semantic security means that, given two different chosen plaintexts $m_0 \neq m_1$, and a ciphertext $c_b = \mathsf{Enc}_{pk}(m_b)$ for some fixed public-key pk and $b \in_{\mathcal{R}} \{0, 1\}$, it should be hard to decide whether c_b encrypts m_0 or m_1. Key privacy means that, given two different public-keys $pk_0 \neq pk_1$, and a ciphertext $c_b = \mathsf{Enc}_{pk_b}(m)$ for some fixed message m and $b \in_{\mathcal{R}} \{0, 1\}$, it should be hard to decide whether c_b has been created by using pk_0 or pk_1.

Use of rerandomizable encryption. Rerandomizable public-key encryption can be used to provide a symmetric authentication key to authorized communication partners (e.g., trusted verifiers) without disclosing the identity of the token to unauthorized entities. Moreover the computations performed by the token are still contained in the more efficient symmetric-key setting.

During creation of a token T, the token issuer encrypts the *token authentication key K_T* of token T with a public encryption key pk_V whose corresponding secret decryption key is known to all verifiers (or their security modules or the trusted server). The resulting ciphertext $c_T = \mathsf{Enc}_{pk_V}(K_T)$ is then stored in the memory of the token. Whenever token T engages a protocol run with a verifier, it first sends its ciphertext c_T. In case the recipient knows the correct decryption key, it can decrypt K_T and use it in a subsequent authentication protocol.

Since an honest verifier must verify that a token has been created by a genuine issuer, a digital signature scheme is used to certify the token authentication key. However, this signature is static data and thus cannot be transmitted to the verifier as plaintext since this would enable tracing of the token and thus violate *location privacy*. Therefore, the signature must be included into the rerandomizable ciphertext.

However, c_T is a static ciphertext and must be frequently rerandomized in order to provide *location privacy*. Therefore, anonymizers must read c_T, rerandomize it to $c'_T \leftarrow \mathsf{Rand}(c_T)$, and replace c_T with c'_T [2]. Since all known rerandomizable encryption schemes require public-key operations (which in turn implies modular exponentiations) to rerandomize a ciphertext, a symmetric-key token cannot rerandomize its ciphertext on its own. Thus, unlinkability relies on the availability of anonymizers that are not controlled by the token. Basically, there are four possibilities to realize anonymizers:

1. *Integrated anonymizers:* The anonymizer may be integrated into the token. This would enable gapless location privacy while improving practicability. However, all known rerandomizable public-key encryption schemes require to compute public-key operations (e.g., exponentiations) in order to rerandomize a ciphertext. Thus, this approach is not applicable to symmetric-key tags.
2. *Public anonymizers:* Anonymizers may be public, which means that they can be constructed and run by everyone. However, public anonymizers as proposed by [2] enable adversaries to put up malicious anonymizers that can perform denial-of-service attacks. Therefore, to fulfill security requirement *availability*, it is necessary that anonymizers are trusted by the users. In return, a trusted anonymizer must authenticate to a token before it is allowed to anonymize it. In practice there may be a variety of public anonymizing service providers the user may choose from the one he trusts.

   Authentication of anonymizers can be realized in the same way as described above for verifiers. Each token may be initialized with an additional rerandomizable ciphertext c_A that encrypts a token-specific symmetric *anonymizer authentication key* K_A under a public-key pk_A whose secret key sk_A is known to all anonymizers trusted to anonymize the specific token. Thus, only trusted anonymizers can decrypt c_A to obtain K_A and use it to authenticate to the token to be anonymized.
3. *Anonymizers controlled by transit enterprise:* Anonymizers may be controlled by the (trusted) transit enterprise. For instance, anonymizers may be included into verifiers or mounted at the stations or in the vehicles of the transit enterprise.
4. *User-controlled anonymizers:* Each user may own an anonymizer that can only be used to rerandomize his own tags. Therefore the user must provide

the public key pk_A of his anonymizer A to the token issuer during the process of issuing an e-ticket.

To summarize, the user must trust the anonymizer to respect his privacy. However, this is a reasonable assumption since the anonymizer is either under his control or managed by a trusted entity (e.g., the transit enterprise).

4.2 Protocol Descriptions

The issue protocol. A user U requests token issuer I to create a token with his desired usage conditions ρ_T (e.g., ticket type, expiration date, geographical usage restrictions, etc.) and therefore provides public-key pk_A of his anonymizer A. Issuer I then creates the token authentication key K_T and anonymizer authentication key K_A for token T. After that, issuer I derives the corresponding helper data (z_T, w_T) and (z_A, w_A) for the PUF of token T as described in Section 4.1. Then, issuer I creates a certificate $\sigma_T = \mathsf{Sign}_{sk_I}(K_T, \rho_T)$ and two rerandomizable ciphertexts $c_T = \mathsf{Enc}_{pk_V}(K_T, \rho_T, \sigma_T)$ and $c_A = \mathsf{Enc}_{pk_A}(K_A)$. Finally, issuer I writes the tuple (w_T, w_A, c_T, c_A) to the (unprotected) memory of token T and physically passes token T to user U.

The anonymize protocol. In order to anonymize a token T, anonymizer A broadcasts an anonymization request. On receipt of this request, token T uses its random number generator to create a random challenge N_T, reads both ciphertexts c_T and c_A from its memory, and sends the tuple (N_T, c_T, c_A) to anonymizer A that then uses the rerandomization function of the rerandomizable encryption scheme to rerandomize both ciphertexts (c_T, c_A) to (c'_T, c'_A). After that, anonymizer A uses its secret decryption key sk_A to decrypt the anonymizer authentication key K_A from ciphertext c_A, uses K_A to authenticate message (K_A, c'_T, c'_A, N_T), which A then sends to token T. On receipt of this message, token T recovers its anonymizer authentication key K_A by reading helper data w_A from its memory and challenging its PUF as described in Section 4.1. If token T can successfully verify the authenticity of tuple (K_A, c'_T, c'_A, N_T) w.r.t. to key K_A, token T updates both ciphertexts (c_T, c_A) stored in its memory to the ciphertexts (c'_T, c'_A) received from anonymizer A. If the authenticity of the response of anonymizer A cannot be verified, token T aborts.

The prove protocol. To verify the authenticity of token T, a verifier V first broadcasts a verification request. On receipt of this request, token T reads ciphertext c_T from its memory and sends it to the verifier V, who then uses its secret decryption key sk_V to decrypt (K_T, ρ_T, σ_T) from c_T. Then, verifier V uses public verification key pk_I of token issuer I to verify σ_T. If verification of σ_T fails or the usage conditions ρ_T associated with token T are violated, verifier V rejects. Otherwise, it continues by using K_T to engage an symmetric-key based authentication protocol with token T. Token T can recover its token authentication key K_T by reading helper data w_T from its memory and challenging its

PUF as described in Section 4.1. Verifier V accepts token T as authentic token if token T successfully completes the authentication protocol w.r.t key K_T. If authentication fails, verifier V rejects token T.

4.3 Analysis of the Framework

This section informally analysis which of the requirements of Section 2.4 are fulfilled by the solution presented in Section 4. We will provide formal proofs in an extended version of the security and privacy model of [30] in a follow-up paper.

Authentication. The solution presented in Section 4 fulfills all of the authentication requirements of Section 2.4:

Authentication: Honest verifiers will only accept tokens whose token authentication key has been certified by a genuine token issuer.

Unforgeability: The properties of the PUF, the underlying authentication protocol, and the semantic security of the rerandomizable encryption scheme ensure that a valid token cannot be cloned or simulated by an adversary since its secrets cannot be extracted. Moreover, the security of the digital signature scheme guarantees that an adversary cannot create valid tokens on his own since he cannot forge signatures.

Availability: The token only updates its internal memory with data that has been authenticated by an authorized (i.e., trusted) anonymizer. Thus, an adversary cannot tamper with the data stored on the token.

Privacy. The solution presented in Section 4 fulfills the following privacy requirements of Section 2.4 w.r.t. to trust model TM 2 (or trust model TM 3 if verifiers are equipped with security modules or are connected to a remote trusted server) as described in Section 4:

Confidentiality: Since the rerandomizable encryption scheme is required to be semantically secure, no information on the secrets of the token is revealed by the corresponding ciphertexts. Moreover, the underlying authentication scheme is required not to disclose any user-related information. Thus, an adversary cannot obtain any information on the token or the user.

Location Privacy: Tokens can be traced between two randomizations. However, if an adversary misses only one rerandomization, he cannot trace a token any more because of the semantic security of the rerandomizable encryption scheme and the properties of the authentication scheme.

Our framework currently does not provide backwards and forward traceability, and we leave this as an interesting open problem.

Functional requirements. The solution presented in Section 4 fulfills all of the functional of Section 2.4:

Efficiency: Verification of a token requires to run a symmetric-key-based authentication protocol between the token and the verifier. Moreover, a single public-key decryption and a single signature verification must be performed by the verifier. This computational effort is comparable to existing schemes currently used in practice (e.g., [1,26]) since the additional operations that must be performed by the verifier can be neglected due to the computing power of currently available RFID readers.
Scalability: The solution does not depend on the number of tokens.

5 Conclusion, Open Problems, and Future Work

Summary of contribution. We analyzed the viability of current proposals for privacy-preserving e-tickets and examined the applicability of privacy-enhancing RFID-based protocols. We showed that existing approaches are not suited for the application scenario of e-tickets and presented a solution based on existing cryptographic tools and current RFID technology.

Open research problems. As discussed in Section 3.2, all currently known privacy-preserving authentication schemes for tokens that are limited to symmetric cryptography seem to require the token issuer to be trusted. Therefore, it would be interesting to find a scheme based on symmetric cryptography but similar to the one that provides similar properties as anonymous credential systems.

Currently, our approach does not provide forward and backward security. Forward and backward-secure anonymous symmetric-key based authentication schemes require frequent update of the secrets of the tokens [18]. However, since secrets are protected by PUFs it is not trivial to update them for both, the token and the verifier, in a way that ensures forward and backwards traceability.

Acknowledgments. This work has been supported in party by the European Commission through the Network of Excellence ECRYPT II.

The work of the third authors has been also supported in part by the European Commission through the FP7 Information Communication Technologies programme, under Contract FET-215270 FRONTS (Foundations of Adaptive Networked Societies of Tiny Artefacts).

References

1. Calypso Networks Association. Web site of Calypso Networks Association. `http://www.calypsonet-asso.org/`, May 2007.
2. Giuseppe Ateniese, Jan Camenisch, and Breno de Medeiros. Untraceable RFID tags via insubvertible encryption. In *Proceedings of the 12th ACM Conference on Computer and Communications Security, November 7–11, 2005, Alexandria, VA, USA*, pages 92–101. ACM Press, 2005.

3. Mihir Bellare, Alexandra Boldyreva, Anand Desai, and David Pointcheval. Key-privacy in public-key encryption. In *7th International Conference on the Theory and Application of Cryptology and Information Security, Gold Coast, Gold Coast, Australia, December 9–13, 2001, Proceedings*, volume 2248 of *LNCS*, pages 566–582. Springer Verlag, 2001.
4. Mihir Bellare, Chanathip Namprempre, and Gregory Neven. Security proofs for identity-based identification and signature schemes. Cryptology ePrint Archive: Report 2004/252, September 2004. Available at `http://eprint.iacr.org/2004/252`.
5. Jan Camenisch, Susan Hohenberger, and Anna Lysyanskaya. Compact e-cash. In *24th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Aarhus, Denmark, May 22–26, 2005, Proceedings*, volume 3494 of *Lecture Notes on Computer Science (LNCS)*, pages 302–321. Springer Verlag, 2005.
6. Jan Camenisch and Anna Lysyanskaya. Signature schemes and anonymous credentials from bilinear maps. In *24th Annual International Cryptology Conference, Santa Barbara, California, USA, August 15–19, 2004, Proceedings*, volume 3152 of *LNCS*, pages 56–72. Springer Verlag, 2004.
7. Nicolas T. Courtois, Karsten Nohl, and Sean O'Neil. Algebraic attacks on the Crypto-1 stream cipher in MiFare classic and oyster cards. Cryptology ePrint Archive, Report 2008/166, 2008. Available at `http://eprint.iacr.org/2008/166/`.
8. Srinivas Devadas, Edward Suh, Sid Paral, Richard Sowell, Tom Ziola, and Vivek Khandelwal. Design and implementation of PUF-based unclonable RFID ICs for anti-counterfeiting and security applications. In *IEEE International Conference on RFID 2008, Las Vegas, NV, USA, 16–17 April, 2008*, pages 58–64. IEEE Computer Society, 2008.
9. Tassos Dimitriou. A lightweight RFID protocol to protect against traceability and cloning attacks. In *Proceedings of the First International Conference on Security and Privacy for Emerging Areas in Communications Networks (SecureComm), September, 05–09, 2005*, pages 59–66. IEEE Computer Society, 2005.
10. Near Field Communication Forum. Web site of Near Field Communication (NFC) Forum. `http://www.nfc-forum.org/`, April 2008.
11. Sony Global. Web site of Sony FeliCa. `http://www.sony.net/Products/felica/`, June 2008.
12. Shafi Goldwasser and Silvio Micali. Probabilistic encryption. *Journal of Computer and System Sciences*, 28:270–299, 1984.
13. Philippe Golle, Markus Jakobsson, Ari Juels, and Paul Syverson. Universal re-encryption for mixnets. In *The Cryptographers' Track at the RSA Conference 2004, San Francisco, CA, USA, February 23–27, 2004, Proceedings*, volume 2964 of *LNCS*, pages 163–178. Springer Verlag, 2004.
14. Dirk Henrici and Paul Müuller. Hash-based enhancement of location privacy for radio-frequency identification devices using varying identifiers. In *Proceedings of the Second IEEE Annual Conference on Pervasive Computing and Communications Workshops, March, 14–17, 2004*, pages 149–153. IEEE Computer Society, 2004.
15. Thomas S. Heydt-Benjamin, Hee-Jin Chae, Benessa Defend, and Kevin Fu. Privacy for public transportation. In *6th International Workshop, PET 2006, Cambridge, UK, June 28–30, 2006, Revised Selected Papers*, volume 4258 of *Lecture Notes on Computer Science (LNCS)*, pages 1–19. Springer Verlag, 2006.
16. Ari Juels. RFID security and privacy: A research survey. *Journal of Selected Areas in Communication (J-SAC)*, 24(2):381–395, February 2006.

17. Ari Juels and Ravikanth Pappu. Squealing euros: Privacy protection in RFID-enabled banknotes. In *7th International Conference, FC 2003, Guadeloupe, French West Indies, January 2003, Revised Papers*, volume 2742 of *LNCS*, pages 103–121. Springer Verlag, 2003.
18. Chae Hoon Lim and Taekyoung Kwon. Strong and robust RFID authentication enabling perfect ownership transfer. In *8th International Conference, ICICS 2006, Raleigh, NC, USA, December 4–7, 2006, Proceedings*, volume 4307 of *LNCS*, pages 1–20. Springer Verlag, 2006.
19. Anna Lysyanskaya. *Signature Schemes and Applications to Cryptographic Protocol Design*. PhD thesis, Massachusetts Institute of Technology, Department of Electrical Engineering and Computer Science, Cambridge, Massachusetts, USA, September 2002.
20. David Molnar and David Wagner. Privacy and security in library RFID: Issues, practices, and architectures. In *Proceedings of the 11th ACM Conference on Computer and Communications Security, Washington, DC, USA, October 25–29, 2004*, pages 210–219. ACM Press, 2004.
21. Karsten Nohl and Henryk Plötz. MiFare — little security despite obscurity. `http://events.ccc.de/congress/2007/Fahrplan/events/2378.en.html`, 2007.
22. Miyako Ohkubo, Koutarou Suzuki, and Shingo Kinoshita. Efficient hash-chain based RFID privacy protection scheme. International Conference on Ubiquitous Computing (UbiComp), Workshop Privacy: Current Status and Future Directions, Nottingham, UK, September, 2004, September 2004.
23. Ronny Wichers Schreur, Peter van Rossum, Flavio Garcia, Wouter Teepe, Jaap-Henk Hoepman, Bart Jacobs, Gerhard de Koning Gans, Roel Verdult, Ruben Muijrers, and Ravindra Kali andVinesh Kali. Security flaw in MiFare Classic. `http://www.sos.cs.ru.nl/applications/rfid/pressrelease.en.html`, March 2008.
24. NXP Semiconductors. Web site of MIFARE. `http://mifare.net/`, May 2007.
25. Boyeon Song and Chris J. Mitchell. RFID authentication protocol for low-cost tags. In *Proceedings of the First ACM Conference on Wireless Network Security, Alexandria, Virginia, USA, March 31 – April 2, 2008.*, pages 140–147. ACM Press, 2008.
26. Spirtech. CALYPSO functional specification: Card application, version 1.3. `http://calypso.spirtech.net/`, October 2005.
27. Gene Tsudik. YA-TRAP: Yet Another Trivial RFID Authentication Protocol. In *Proceedings of the 4th Annual IEEE International Conference on Pervasive Computing and Communications Workshops, March 13–17, 2006*, volume 2802 of *LNCS*, pages 640–643. IEEE Computer Society, 2006.
28. Pim Tuyls and Lejla Batina. RFID-tags for anti-counterfeiting. In *The Cryptographers' Track at the RSA Conference 2006, San Jose, CA, USA, February 13–17, 2005, Proceedings*, volume 3860 of *Lecture Notes on Computer Science (LNCS)*, pages 115–131. Springer Verlag, 2006.
29. Pim Tuyls, Boris Škoriç, and Tom Kevenaar, editors. *Security with Noisy Data — On Private Biometrics, Secure Key Storage, and Anti-Counterfeiting.* Springer-Verlag, 2007.
30. Serge Vaudenay. On privacy models for RFID. In *13th International Conference on the Theory and Application of Cryptology and Information Security, Kuching, Malaysia, December 2–6, 2007, Proceedings*, volume 4833 of *LNCS*, pages 68–87. Springer Verlag, 2007.
31. Stephen A. Weis, Sanjay E. Sarma, Ronald L. Rivest, and Daniel W. Engels. Security and privacy aspects of low-cost radio frequency identification systems.

In *First International Conference on Security in Pervasive Computing, Boppard, Germany, March 12–14, 2003, Revised Papers*, volume 2802 of *LNCS*, pages 50–59. Springer Verlag, 2003.

The Curupira-2 Block Cipher for Constrained Platforms: Specification and Benchmarking

Marcos Simplício Jr[1], Paulo S. L. M. Barreto[1] *, Tereza C. M. B. Carvalho[1], Cintia B Margi[1], and Mats Näslund[2]

[1]Laboratory of Computer Architecture and Networks, PCS-EP, University of São Paulo, Brazil
{mjunior,pbarreto,carvalho,cbmargi}@larc.usp.br
[2]Ericsson Research - Stockholm, Sweden
mats.naslund@ericsson.com

Abstract. Privacy is a key concern in Location Based Applications (LBAs), especially due to their intensive use resource constrained devices in which general purpose ciphers are difficult to deploy. In this paper, we address this issue by specifying a new, faster key-schedule algorithm for the Curupira block cipher. This special-purpose cipher follows the Wide Trail Strategy (such as AES) and is tailored for resource-constrained platforms, such as sensors and mobile devices. Furthermore, we present our benchmark results for both the Curupira-1 (which adopts the original key-schedule specification) and the Curupira-2 (which adopts the new one) in appropriate testbeds.

Keywords: location based services, symmetric cryptography, involutional block ciphers, sensor networks, constrained platforms, ciphers benchmark.

1 Introduction

The continuous and widespread development of context-aware applications based on Wireless Sensor Networks (WSNs) shows their potential to allow a high level of integration between computers and the physical environment. Location-Based Applications (LBAs) play an important role in this scenario, automatically adapting their behaviors to the available geo-spatial location information and the nearby sensors and devices. This way, these systems are able to provide both novel and more effective services. However, due to the sensitive nature of the data involved (both location information and other data collected by the network nodes) the security of the communication in these applications is an important concern.

Battery-powered sensor nodes and other limited platforms normally employed in LBAs impose several constraints over the cryptographic algorithms

* Supported by the Brazilian National Council for Scientific and Technological Development (CNPq) under grant 312005/2006-7. †This work was supported by the Research and Development Centre, Ericsson. Telecomunicações S.A., Brazil

that can be effectively deployed. For example, commercial motes usually have a memory size of 8-12 KB for code and 512-4096 bytes of RAM, as well as 4-16 MHz processors [21, 17]. Moreover, messages exchanged in these applications are frequently small, a typical packet being 24 bytes in length [22, 30]. In this context, complex all-purpose algorithms not only take longer to run but also consume more energy, which motivates the research for more efficient alternatives.

To date, many architectures have been proposed to provide security in WSNs. One of the most popular is TinySec [19], which offers link layer security to TinyOS [18], the *de facto* standard operating system for sensor networks. As default block cipher, TinySec has chosen Skipjack [33] due to its superior performance. Meanwhile, RC5 [34] was not considered as an alternative since it is considered to be encumbered with patents and, even if it can run faster than Skipjack when the round keys are pre-computed, this incurs extra RAM requirements [15]. However, as Skipjack uses relatively small (80-bit) keys and 31 out of its 32 rounds can be successfully cryptanalyzed [9], it presents a very low margin of security. These observations lead to security concerns regarding TinySec, as well as other architectures based on the same cipher, such as TinyKeyMan [24], MiniSec [25] and Sensec [23].

The literature includes numerous analyses of modern cryptographic algorithms, aiming to identify those well-suited to WSNs both in terms of security and performance. One of the most extensive is presented by [21], which concludes that Skipjack is the most energy-efficient of all surveyed ciphers, while the Advanced Encryption Standard (AES) [32] and MISTY1 [27] are considered reasonable alternatives in scenarios with higher security requirements. However, MISTY1 is considered to be encumbered with patents, while AES has larger code, memory, and energy requirements, as well as a block size which is too big for sensor networks. It could also be possible to rely on cryptographic hardware [16], but this is not a solution available in many modern devices.

An alternative to address these issues is the adoption of CURUPIRA [6], a special-purpose block cipher specially developed with constrained platforms in mind. The cipher follows the Wide Trail strategy [11], such as the AES itself, which assures a good security against cryptanalysis. Also, it presents an involutional structure (meaning that the encryption and decryption processes are identical except by the key-schedule) and is very flexible in terms of implementation. This way, the cipher is well adapted to resource constrained scenarios such as those faced by LBAs.

In this paper, we propose a new, faster key-schedule algorithm for the CURUPIRA algorithm and analyze its security. We also present a benchmark comparing Skipjack, AES and CURUPIRA in relevant platforms. In order to discern between the two versions of the CURUPIRA cipher, we write 'CURUPIRA-1" for the one adopting the original key-schedule and "CURUPIRA-2" for the new specification. We write simply "CURUPIRA" when the discussion applies to both.

This document is organized as follows. We introduce basic mathematical tools and notation in section 2. An overview of the CURUPIRA-1, including its key-schedule algorithm, is given in section 3. The second version of the key-schedule is presented in section 4, which also discuss security, performance and implementation issues for the resultant CURUPIRA-2 block cipher. Our benchmark results are presented in section 5. We conclude in section 6.

2 Mathematical preliminaries and notation

The finite field $\mathrm{GF}(2^n)$ will be represented as $\mathrm{GF}(2)[x]/p_n(x)$, where $p_n(x)$ is a primitive pentanomial of degree n over $\mathrm{GF}(2)$, i.e. $\deg(p_n(x)) \equiv n$. This way, all multiplications over $\mathrm{GF}(2^8)$ are made modulo $p_8(x) = x^8 + x^6 + x^3 + x^2 + 1$. This choice of $p_8(x)$ incurs in a simple form for the primitive cube root of unity, $c(x) = x^{85} \bmod p_8(x) = x^4 + x^3 + x^2$.

An element $u = u_7x^7 + \ldots + u_ix^i + \ldots + u_0$ of $\mathrm{GF}(2^8)$ where $u_i \in \mathrm{GF}(2)$ for all $i = 0, \ldots, 7$, will be denoted by the numerical value $u_7 \cdot 2^7 + \ldots + u_i \cdot 2^i + \ldots + u_0$, written in hexadecimal notation. Thus, the polynomial $c(x)$ is written `1C`, while $p_8(x)$ is written `14D`. The multiplication by the polynomials x and $c(x)$ are denoted `xtimes` and `ctimes`, respectively.

The set of all $3 \times n$ matrices over $\mathrm{GF}(2^m)$ is denoted by $\mathbb{M}_n$. Let D and E denote the MDS matrices (see [26] for a definition) as follows:

$$D = \begin{bmatrix} 3 & 2 & 2 \\ 4 & 5 & 4 \\ 6 & 6 & 7 \end{bmatrix}, \; E = \begin{bmatrix} 1{+}c(x) & c(x) & c(x) \\ c(x) & 1{+}c(x) & c(x) \\ c(x) & c(x) & 1{+}c(x) \end{bmatrix}.$$

3 Overview of the CURUPIRA-1 structure

This section gives a brief description of the CURUPIRA-1 original specification. For further details, we refer to [6].

The CURUPIRA is a block cipher specially tailored for constrained platforms. It operates on 96-bit data blocks (organized as $\mathbb{M}_4$ matrices, mapped by columns instead of by rows) and accepts 96-, 144- or 192-bit keys, with a variable number of rounds. The cipher round structure is similar to the one in BKSQ [12], with the advantage of being involutional, resulting in a more compact cipher. Its round function structure is used for both CURUPIRA-1 and CURUPIRA-2 and is composed by the following self-inverse transforms (see Figure 1:

- *Nonlinear Layer* (γ): all bytes in the block pass through a highly nonlinear S-Box, identical to that used in ANUBIS [7] and KHAZAD [8] block ciphers;
- *Linear Diffusion Layer*(θ): the block is left-multiplied by the MDS matrix D (the one defined in section 2), which results in intra-columnar diffusion;
- *Permutation Layer* (π): all the bytes in the second and third rows of the block are permuted according to the rule $\pi(a) = b \Leftrightarrow b_{i,j} = a_{i,i \oplus j}$, $0 \leqslant i < 3$, $0 \leqslant j < n$;
- *Key addition Layer* (σ): the round key is XORed with the block.

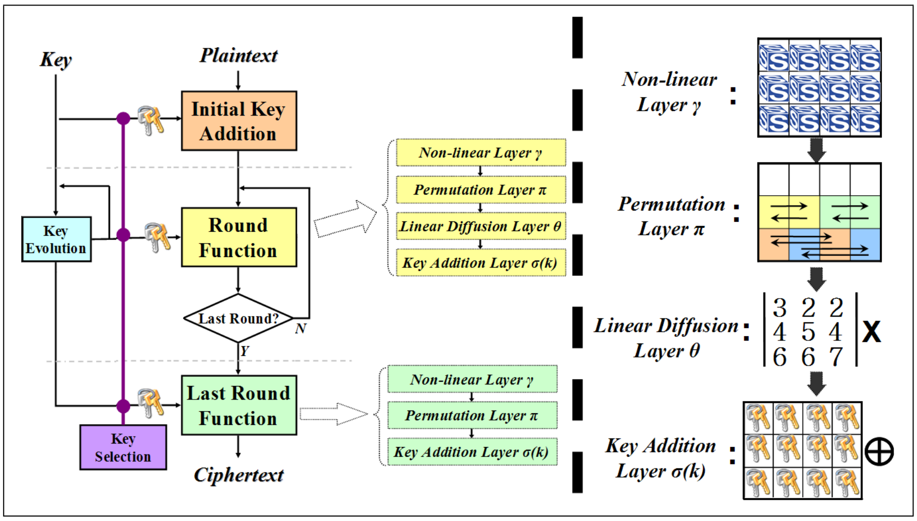


Fig. 1. Curupira Round Structure

These transforms only involve basic operations such as table lookups, XORs and byte shifts. Thus, they can be implemented in most platforms in a very efficient way. Nonetheless, when space is available, they can be further accelerated using pre-computed tables, operating over entire columns of the block instead of byte-to-byte.

3.1 The Curupira-1 key-schedule

The Curupira-1 key-schedule algorithm is easily invertible and follows a structure closely related to the one dictated by the Wide Trail Strategy, which assures a high diffusion speed. Also, it has the advantage of being cyclical, which means that the original key is recovered after a certain number of rounds, avoiding the need of storing any intermediary sub-key during both encryption and decryption.

In this first construction, a $48t$-bit user key $\mathcal{K}$, $2 \leqslant t \leqslant 4$ is internally represented as a matrix $K \in \mathbb{M}_{2t}$. To generate a sub-key K^{n+1} from its predecessor K^n, the sub-keys pass through three different transformations (illustrated in Figure 2):

- Constant Addition ($\sigma(q)$): a set of nonlinear constants, incrementally taken from the S-Box, are XORed with the bytes in the *first row* of the round key; this way, for a $48t$-bit key, the r^{th} round and the column j, the $q_j^{(r)}$ constant is given by: $q_j^{(0)} = 0$ and $q_j^{(r)} = S[2t(r-1)+j]$, $0 \leqslant j \leqslant 2t$;
- Cyclic Shift (ξ): rotates the second row one position to the left, and rotates the third row one position to the right, keeping the first row unchanged;
- Linear Diffusion (μ): the round-key is left-multiplied by the matrix E (defined in section 2).

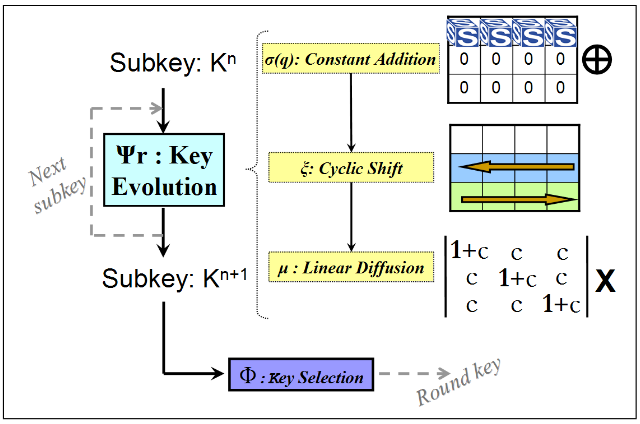


Fig. 2. The original key-schedule specification (adopted by Curupira-1)

Furthermore, the round keys $\kappa^{(r)}$ effectively combined with the data blocks are chosen by the Key Selection algorithm ϕ_r, which applies the S-Box to the first row of the sub-key K^r and truncates it to 96 bits (i.e., the size of the block). Thus, even if the Key Selection is not part of the key evolution, it adds nonlinearity to the key-schedule.

4 The Curupira-2 key-schedule

The Curupira-1 key-schedule specification is quite conservative, aiming for a high security level against related-key attacks. Thus, it is reasonable to adopt a simpler yet secure key-schedule algorithm in order to improve the overall cipher performance. That is the approach of the Curupira-2 key-schedule, which will be described in the following sections.

4.1 Key representation

A $48t$-bits long user key $\mathcal{K}$ $(2 \leqslant t \leqslant 4)$ is internally represented as an element K belonging to the finite field $\mathrm{GF}(2^{48t}) = \mathrm{GF}(2)/p_{48t}(x)$, where $p_{48t}(x)$ is a pentanomial in GF(2) chosen in such a way that x^8 is a primitive root of $p(x)$. An element $u(x)$ in this field can be seen as a byte vector, i.e. $u = (U_{6t-1}, \ldots, U_0)$, where U_0 indicates the lesser significant byte. This way, the cryptographic key $\mathcal{K}$ is directly mapped to K, starting at its most significant byte, K_{6t-1}.

A pentanomial representation was chosen because primitive pentanomials are available for all values of t used by the Curupira. In fact, the use of trinomials would result in a better performance, but unfortunately they do not exist for fields of type $\mathrm{GF}(2^n)$ when n is multiple of 8 [28] and, thus, they cannot be used for any of the cipher key sizes.

For reasons that will become clearer in section 4.3, the pentanomials chosen for Curupira-2 are:

- $p_{96}(x) = x^{96} + x^{16} + x^{13} + x^{11} + 1$;
- $p_{144}(x) = x^{144} + x^{56} + x^{53} + x^{51} + 1$;
- $p_{192}(x) = x^{192} + x^{43} + x^{41} + x^{40} + 1$.

4.2 Schedule constants

The *schedule constants* are denoted $q^{(s)}$, where the index ${}^{(s)}$ indicates the round in which they are applied. As in the CURUPIRA-1, the constants are directly taken from the S-box and, thus, no extra storage is needed. This time, however, they are interpreted as elements of $\mathrm{GF}(2^{48t})$ in such a way that $q^{(0)} = 0$ and, for $s > 0$, $q^{(s)} = (S[s-1], 0 \ldots, 0)$ i.e. a single S-box output is mapped to the most significant byte of $q^{(s)}$. As shown in the next section, this is a strategic position that makes each constant affect exactly 3 bytes of the round key right after its addition.

4.3 The key evolution Υ_s

The sub-keys are updated during the cipher operation by means of two operations: a reversible transform, $\aleph : \mathrm{GF}(2^{48t}) \to \mathrm{GF}(2^{48t})$; and an auto-inverse transform $\eta : \mathrm{GF}(2^{48t}) \to \mathrm{GF}(2^{48t})$. They are defined in such a way that $\aleph(u) \equiv u \cdot x^8$ and, for the polynomials $u = (U_{6t-1}, \ldots, U_0)$ and $v = (V_{6t-1}, \ldots, V_0)$ in $\mathrm{GF}(2^{48t})$:

$$v = \eta(u) \Leftrightarrow \begin{cases} V_i = U_{11-i} \oplus U_{12+i} & \text{if } 0 \leqslant i < 6t - 12; \\ V_i = U_i & \text{otherwise.} \end{cases}$$

Together with the schedule constants, these transforms compose the *key evolution function* $\Upsilon_r : \mathrm{GF}(2^{48t}) \to \mathrm{GF}(2^{48t})$, defined as $\Upsilon_r(u) \equiv \eta \circ \aleph(u \oplus q^{(r)})$, in such a way that $K^{(0)} \equiv K$ and $K^{(r+1)} = \Upsilon_r(K^{(r)})$.

The η transform is used to ensure a greater diffusion to the schedule when keys greater than 96 bits are adopted, since it combines some of the least significant bytes of the key with the most significant ones. Without this operation, two 144-bit keys differing only at the byte K_{11} would generate sub-keys whose 12 least significant bytes would be identical for the first 6 rounds; also, for 192-bit keys, this result would hold true for the first 12 sub-keys. As these least significant bytes are the ones actually selected in each round, the effect of the η transform is essential to assure a higher diffusion speed for 144- and 192-bit keys.

Also, the $\aleph$ transform is particularly interesting due to its performance, specially on resource-constrained platforms, as stated in the following theorem:

Theorem 1. *Let $p(x) = x^n + x^{k_3} + x^{k_2} + x^{k_1} + 1$ be a primitive pentanomial of degree $n = bw$ over $\mathrm{GF}(2)$ such that $k_3 > k_2 > k_1$, $k_3 - k_1 \leqslant w$, and either $k_3 \bmod w = 0$ or $k_1 \bmod w = 0$. Then multiplication by x^w in $\mathrm{GF}(2^n) = \mathrm{GF}(2)[x]/p(x)$ can be implemented with no more than 5 XORs and 4 shifts on w-bit words. Moreover, if 2×2^w bytes of storage are available, the cost drops to no more than 2 XORs on w-bit words and 2 table lookups.*

Proof. For $u = \bigoplus_{d=0}^{n-1} (u_d x^d) \in \mathrm{GF}(2^n)$, let $U_i = u_{wi+w-1}x^{w-1} + \ldots + u_{wi}$ where $i = 0, \ldots, b-1$, so that $u = U_{b-1}x^{w(b-1)} + U_{b-2}x^{w(b-2)} + \ldots + U_0$, which for brevity we write $u = (U_{b-1}, \ldots, U_0)$. Then one can compute $u \cdot x^w$ as:

$$\begin{aligned}
&(U_{b-1}x^{w(b-1)} + U_{b-2}x^{w(b-2)} + \ldots + U_0) \cdot x^w = \\
&U_{b-1}x^n + U_{b-2}x^{w(b-1)} + \ldots + U_0 x^w = \\
&U_{b-2}x^{w(b-1)} + \ldots + U_0 x^w + U_{b-1}(x^{k_3} + x^{k_2} + x^{k_1} + 1) = \\
&(U_{b-2}, \ldots, U_0, U_{b-1}) \oplus U_{b-1}(x^{k_3} + x^{k_2} + x^{k_1}).
\end{aligned}$$

Assume that $k_3 = w(k+1)$ for some k; the case $k_1 \bmod w = 0$ is handled analogously. Thus:

$$u \cdot x^w = (U_{b-2}, \ldots, U_0, U_{b-1}) \oplus U_{b-1}(x^w + x^{w-k_3+k_2} + x^{w-k_3+k_1})x^{wk}$$

Since $\deg(U_{b-1}) \leqslant w - 1$ and $\deg(x^w + x^{w-k_3+k_2} + x^{w-k_3+k_1}) = w$, their product is a polynomial of degree not exceeding $2w - 1$, and hence it fits two w-bit words for any value of U_{b-1}. Besides, multiplication of this value by x^{wk} corresponds to simply displacing it k words to the left. We can define:

$$\begin{aligned}
&T_1[U] \equiv U \oplus (U \gg (w - k_3 + k_2))) \oplus (U \gg (w - k_3 + k_1))), \\
&T_0[U] \equiv (U \ll (k_3 - k_2)) \oplus (U \ll (k_3 - k_1));
\end{aligned}$$

Thus, we can write $u \cdot x^w = (U_{b-2}, \ldots, U_k \oplus T_1[U_{b-1}], U_{k-1} \oplus T_0[U_{b-1}], \ldots, U_0, U_{b-1})$. The values T_1 and T_0 can be either computed on demand or else pre-computed and stored in two 2^w-entry tables. One easily sees by direct inspection that the computational cost is that stated by the theorem.

Applying this theorem for the polynomials adopted by the CURUPIRA-2, we can evaluate the cost of the transforms $\aleph$ and its inverse:

$$\begin{aligned}
&p_{96}(x) = x^{96} + x^{16} + x^{13} + x^{11} + 1 : \\
&\quad \aleph : (U_{11}, \ldots, U_0) \cdot x^8 \quad\;\; = (U_{10}, \ldots, U_1 \oplus T_1[U_{11}], U_0 \oplus T_0[U_{11}], U_{11}); \\
&\quad \aleph^{-1} : (U_{11}, \ldots, U_0) \cdot x^{-8} = (U_0, U_{11}, \ldots, U_2 \oplus T_1[U_0], U_1 \oplus T_0[U_0]);
\end{aligned}$$

$$\begin{aligned}
&p_{144}(x) = x^{144} + x^{56} + x^{53} + x^{51} + 1 : \\
&\quad \aleph : (U_{17}, \ldots, U_0) \cdot x^8 \quad\;\; = (U_{16}, \ldots, U_6 \oplus T_1[U_{17}], U_5 \oplus T_0[U_{17}], \ldots, U_0, U_{17}); \\
&\quad \aleph^{-1} : (U_{17}, \ldots, U_0) \cdot x^{-8} = (U_0, U_{17}, \ldots, U_7 \oplus T_1[U_0], U_6 \oplus T_0[U_0], \ldots, U_1);
\end{aligned}$$

$$\begin{aligned}
&p_{192}(x) = x^{192} + x^{48} + x^{45} + x^{43} + 1 : \\
&\quad \aleph : (U_{23}, \ldots, U_0) \cdot x^8 \quad\;\; = (U_{22}, \ldots, U_5 \oplus T_1[U_{23}], U4 \oplus T_0[U_{23}], \ldots, U_0, U_{23}). \\
&\quad \aleph^{-1} : (U_{23}, \ldots, U_0) \cdot x^{-8} = (U_0, U_{23}, \ldots, U_6 \oplus T_1[U_0], U_5 \oplus T_0[U_0], \ldots, U_1).
\end{aligned}$$

For all key sizes, we have $T_0 = U \oplus (U \gg 5) \oplus (U \gg 3)$ and $T_1 = (U \ll 3) \oplus (U \ll 5)$.

These equations show that both $\aleph$ and $\aleph^{-1}$ transforms have the same cost and, thus, it is also valid for Υ_r and its inverse, $\Upsilon_r^{-1}(u) \equiv (\aleph^{-1} \circ \eta(u)) \oplus q^{(r)}$. In contrast, when compared to the key-schedule of the CURUPIRA-1, this second schedule algorithm has one disadvantage: there is no simple way to reinitialize the key after a reduced number of rounds. However, in many applications, its higher speed both during encryption and decryption can be a much more interesting feature, compensating its lack of cyclicity.

4.4 The key selection ϕ_r^*

The round keys $\kappa^{(r)} \in \mathbb{M}_n$ effectively used in each round are calculated by means of the *key selection function* $\phi_r^* : \mathrm{GF}(2^{48t}) \to \mathbb{M}_n$, defined in such a way that:

$$\kappa^{(r)} = \phi_r^*(K) \Leftrightarrow \begin{cases} \kappa_{i,j}^{(r)} = S[K_{i+3j}^{(r)}] \text{ if } i = 0; \\ \kappa_{i,j}^{(r)} = K_{i+3j}^{(r)} \text{ otherwise.} \end{cases}$$

This way, only the 12 least significant bytes are taken by ϕ_r^*. Also, the S-box is applied to the bytes that will be combined with the first row of the block, adding nonlinearity to the key-schedule, while the bytes for the other rows are taken directly. The whole process involved in this second version of the key-schedule algorithm is depicted in Figure 3.

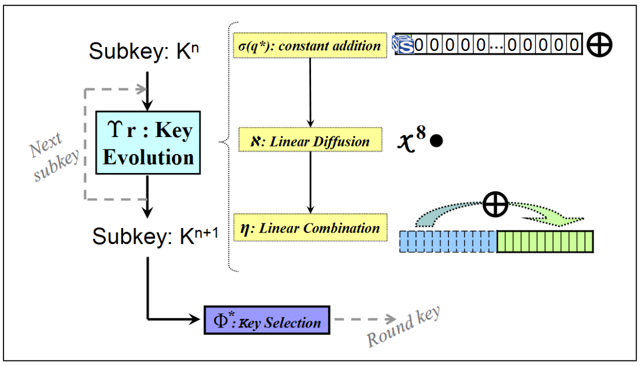


Fig. 3. The new key-schedule specification (adopted by CURUPIRA-2)

4.5 Security analysis revisited

Since, for both versions of our cipher, the round structure remains the same, most of the CURUPIRA-1 security analysis [6, section 4] also applies to the CURUPIRA-2: the adoption of the Wide Trail Strategy in combination with a highly nonlinear S-Box thwarts the most well known modalities of attacks, such as linear, differential and integral cryptanalysis. As a consequence, no attack faster than exhaustive search was found for more than 7 rounds of the cipher. These results were also confirmed by third party analysis [31]. The main difference between the two analysis concerns the existence of weak keys and the viability of related-key attacks.

Weak keys are keys that result in a block cipher mapping with detectable weaknesses, which normally occurs when the nonlinear operations depend on the actual key value. This is not the case for the CURUPIRA, where keys are applied using XOR and all nonlinearity is fixed in the S-box. Also, the nonlinear round constants considerably reduce the probability of fixed points in the key-schedule process, making the existence of weak keys very unlikely.

Related-key attacks exploit a known relationship between different unknown keys, leading to a predictable behavior for the sub-keys generated by the key schedule. Some of the most widespread techniques involve key differentials and key rotations (cf. [10]) in order. Due to its slower diffusion, it is clear that it is easier to find relationships between subsequent sub-keys in CURUPIRA-2 than in CURUPIRA-1, making the former less resistant to related-key attacks. In fact, between any two rounds, the difference in a single internal byte (i.e. in a position that will only be shifted as a result of the multiplication by x^8) results in a difference on a single byte of the next sub-key, which could be somehow exploited. In spite of this, some fundamental elements are introduced to the key-schedule proposed in this paper in order to prevent attacks. First, the nonlinearity introduced by the key selection thwarts related-key attacks involving differentials. Second, the generation of sub-keys does not involve simple rotations, but rather a multiplication over $\mathrm{GF}(2^{48t})$ after the addition of nonlinear constants. Third, the truncation of the sub-keys make some advanced related-key attack variants such as that described in [13, section 4] improbable. Finally, the slow diffusion in the key schedule is counterbalanced by the round function fast diffusion, assuring that each byte of the key affects many block bytes after a few rounds. Together, these features make this kind of attacks unlikely to work against the full cipher.

Furthermore, for key lengths that are larger than the length of one round key, the existence of sets of keys that produce identical values for at least one round key is inevitable. Thus, even if the η transform adds diffusion power to the key-schedule and prevents the existence of trivial sets with this property, they should be more easy to find than in the CURUPIRA-1. Even so, it remains unclear how such keys could possibly be successfully used in a related-key attack.

As a last remark, the CURUPIRA structure involves only simple operations such as XORs, shifts and table lookups. As long as the running times for these transforms are not data-dependent on the target platform, the cipher implementation can avoid many side-channel attacks (such as timing-attacks [20]) in a straightforward way.

4.6 Implementation and Performance Issues

The CURUPIRA-2 algorithm is very flexible in terms of implementation, offering many memory/performance trade-offs. It cannot only use the same techniques developed for the CURUPIRA-1 round functions [6, section 5]. These include the usage of a few tables with pre-computed results, but its key-schedule also allow some useful optimizations depending on resources available.

The round keys can be either computed on-demand or fully pre-calculated and stored in a table for ready access. In the first case, the cipher requires a reduced amount of RAM memory, since only one sub-key is stored at any given time. However, as the key-schedule of CURUPIRA-2 is not cyclic, there is no easy way to compute the first round key from the last one. An easy way to overcome this problem is to use two arrays k_a and k_b to store the first and the last sub-keys, respectively: when one wants to encrypt, it suffices to copy k_a into k_b and reuses k_a memory space to create the encryption sub-keys; in the end, k_a will have the last key while k_b will store the first one, assuring that both sub-keys are always available. The decryption is handled analogously. In this case, the last sub-key could be computed during the cipher initialization. We note that this strategy is only possible because the key schedule is easily invertible.

For 6t-byte keys ($2 \leqslant t \leqslant 4$), the round sub-keys can be calculated in any direction at the cost of one circular permutation, 2+6(t-2) XORs and one computation of T_0 and T_1. Also, T_0 and T_1 can be either implemented using two 256-bit tables or calculated on-the-fly, taking 1 XOR + 2 shifts and 2 XORs + 2 shifts, respectively. In fact, the circular permutation does not need to be effectively implemented: the same effect can be achieved if the index corresponding to the most significant byte of the key is stored and used as the first byte of the key for every calculation; this way, it suffices to update this index after each invocation of the $\aleph$ and $\aleph^{-1}$ operations.

Reviewing the CURUPIRA preliminary calculations [6, section 5.1], the cost of its round function is $3R - 1$ XORs, $2(R - 1)/3$ `xtimes` operations and R S-box lookups per byte. When the key-schedule and key selection are taken into account, we add at most 1/3 S-Box lookups, 1/3 `ctimes` operations and 2 XORs per key byte and per round in the CURUPIRA-1, while this cost drops to at most 5/12 S-Box lookups, 5/8 XORs and 1/12 computations of T_0 and T_1 per key byte and per round in the CURUPIRA-2. In comparison, Skipjack takes basically 48 XORs and 16 F-table lookups per encrypted byte. Thus, supposing that the cost of any of these basic operations are approximately the same and not counting auxiliary operations not directly related to the ciphers structures (such as counter increments and key index updates), CURUPIRA-1 with 96-bit keys and 10 rounds is about $(45 + 27)/64 \approx 112.5\%$ as Skipjack when the round keys computed on demand. On the other hand, CURUPIRA-2 with the same key-size corresponds to $(45 + 7.5)/64 \approx 82\%$ of Skipjack computation in the same conditions. This result should be expected for similar implementations of both ciphers on byte-oriented platforms.

Furthermore, more powerful processors (32-bit servers, for example) could implement the Υ_s transform in a more efficient way, operating over columns instead of bytes. Also, the multiplication by x^8 can be easily implemented using a single table that calculates T_0 and T_1 at the same time, an approach similar to those adopted in some very optimized versions of AES [14].

5 Benchmark

In this section, we present the results of our comparison between CURUPIRA, Skipjack and AES in terms of processing time and memory usage. As discussed in section 1, the motivation behind the choice of Skipjack resides in the results presented in [35] and in [21] which shows that, in spite of its low security level, the cipher is a very interesting choice to achieve a high performance in constrained platforms, surpassing many other hardware-oriented ciphers like MISTY1 and Kasumi [1]. AES, on the other hand, provides higher security but is recommended for less constrained platforms, since it is a less memory-efficient cipher. Considering these remarks, we decided to develop a deep analysis on the comparison of Skipjack and CURUPIRA in both constrained and powerful platforms, while AES is taken into account only on powerful ones. Three different platforms were chosen as testbeds:

– Microcontroller (8 bits): a RISC microcontroller PIC18F8490 [29] equipped with a 8MHz processor, 768 bytes of RAM and a memory size of 16KB for code. The reason behind this choice resides in its capacity, slightly superior to the one presented by the ATmega8535 [2]. This last device, with a 4 MHz processor, 8 KB of flash memory and 512 bytes of RAM, is the one used in the Smart Dust Project [17] for sensor networks.
– Microcontroller Simulator (8 bits): Avrora version 1.6.0 - Beta [36], simulating a microcontroller from the ATmega128 [3] series. The goal of using this simulator is mainly to validate the results obtained with the PIC processor in a more powerful, yet tiny platform.
– Pentium 4 (32 bits): a notebook equipped with Pentium 4 (3.2GHz) and 1GB of RAM. This platform was chosen to evaluate the proposed optimizations of the cipher when the resources in the target platform are abundant.

5.1 Implementation Characteristics

For the 8-bit versions of CURUPIRA and Skipjack, the same C-written implementations were analyzed in both the PIC18F84908 and the Avrora simulator. Furthermore, they adopt similar interfaces in each of these platforms, in order to assure a fair comparison. They also are more speed-oriented than memory-oriented, since the consumption of energy with processing is proportional to the number of operations performed by the algorithm, and this is normally considered the most critical resource in constrained platforms, particularly in WSNs.

For the implementation running on Avrora, as recommended by its documentation, we adopted *avr-objdump* and *avr-gcc* (both GNU utilities) as compilation tools, while *MPLAB IDE v7.60* and *MPLAB C18* compiler are used together with the PIC microcontroller. The speed-optimized versions of each cipher, resulting from the available compiler optimizations, are the ones considered in this document. It is important to notice that, even if both platforms include indirect addressing in their instruction sets, our tests showed that the compilers were not able to fully take advantage of these instructions, resulting in less than optimal machine codes when pointers and/or matrices were used. That is the reason why we decided to evaluate two different programming techniques: one that uses pointers and matrices and another that uses basic-type variables more intensively, avoiding indirect addressing. While the first approach normally results in more flexible code (where the size of the keys can be more easily changed, for example), the second allows more optimized implementations with fixed-size keys (enabling loop unrolling with little loss of compactness)

The implementations running on the 8-bit platforms are detailed below:

Curupira (8 bits) using the proposed optimizations for constrained platforms, we elected two versions of the Curupira for each scheduling algorithm:

1. Curupira$_c$-1: complete version (meaning that it accepts all key sizes) of the Curupira-1. It requires two 256-byte tables, one for the S-Box and another for the `ctimes` operation and uses many pointers and matrices
2. Curupira$_c$-2: complete version of the Curupira-2, using two 256-byte tables for the S-Box and `xtimes` operations. Such as Curupira$_c$-1, it is also based on indirect addressing instructions.
3. Curupira$_{k^{96}}$-1: Curupira-1 restricted to 96-bit keys. It uses the same tables as the Curupira$_c$-1, but relies on basic types instead of indirect addressing instructions.
4. Curupira$_{k^{96}}$-2: Curupira-2 restricted to 96-bit keys, using the same tables as the Curupira$_c$-2 but relying on basic-type variables.

Skipjack (8 bits) two versions were developed according to the specification:

1. Skipjack$_c$: relies on indirect instructions just like Curupira$_c$-1 and Curupira$_c$-2, providing a useful source of comparison with these ciphers. It uses a single 256-byte F-table and calculates the round keys on demand.
2. Skipjack$_k$: adopts programming strategies similar to those present in the Curupira$_{k^{96}}$, strongly relying on operations over basic-type variables instead of matrices and pointers. Such as the Skipjack$_c$, it also uses a 256-byte F-table and calculates the round keys on-the-fly.

For the 32-bits platform, we decided to take advantage of some highly optimized cipher implementations publicly available. The chosen algorithms, written in Java, were compiled and run on Netbeans IDE 5.5, using the JDK 1.6. The details of the implementations are given below:

AES (32 bits) we adopted the implementation of Barreto [5], which pre-computes the round keys and employs ten 256-word tables to greatly accelerate the cipher operation.

Curupira (32 bits) the optimizations in the algorithm are similar to those present in AES, specially regarding the pre-computation of the round keys and the intensive use of tables, in the same number as AES.

Skipjack (32 bits) the cipher tested is a Java adaptation of Barreto's algorithm [4], originally developed in C language. It operates over 16-bit words and stores some important key-dependent pre-computed values in a 10x256-word matrix; this last operation can be seen as a kind of "key-schedule", since it must be performed each time the cipher key is changed.

5.2 Results: 8-bits platforms

The ciphers memory usage, for both 8-bit platforms, is presented in Table 1. This table shows that all tested versions of the Curupira take more space in memory than Skipjack, an expected result considering the higher complexity of its round function and key-schedule algorithms. Despite this difference, the tested ciphers

Table 1. Memory Occupation (in bytes) of the tested ciphers on the 8 bits platforms

Algorithm	*ROM*	*Code-PIC18F8490*	*Code-Avrora Simulator*
Curupira$_c$-1	822	1444	1648
Curupira$_c$-2	512	1238	1718
Curupira$_{k^{96}}$-1	768	1372	1936
Curupira$_{k^{96}}$-2	512	1532	1846
Skipjack$_c$	256	1012	940
Skipjack$_k$	256	972	1352

are both compact enough to be easily deployed in most constrained platforms, taking less than 3KB as a whole.

Both Curupira and Skipjack do not need to pre-compute the round keys and, thus, they require a reduced amount of RAM. We were not able, however, to directly measure the RAM usage with the tools available for the tested platforms. Nevertheless, due to its greater block and key size, we speculate that Curupira takes a higher amount of RAM than Skipjack. For example, when using two arrays to store the first and the last keys, Curupira-2 would take about twice $((96+2*96)/(64+80))$ the amount of RAM needed by Skipjack. These numbers remain very tiny when compared to pre-computed keys storage, though.

For the PIC microcontroller, Figure 4 shows the number of CPU cycles, per byte encrypted, of both tested ciphers. The time measured corresponds to a single encryption of random blocks using different keys. Even if the Curupira$_c$ implementations allow three different key sizes, only the 96-bit keys are depicted in this graph. Also, we explicitly distinguish the processing time required for the key scheduling and the encryption itself (as Skipjack reuses the original key cyclically, we considered it part of the encryption instead of a "key-schedule").

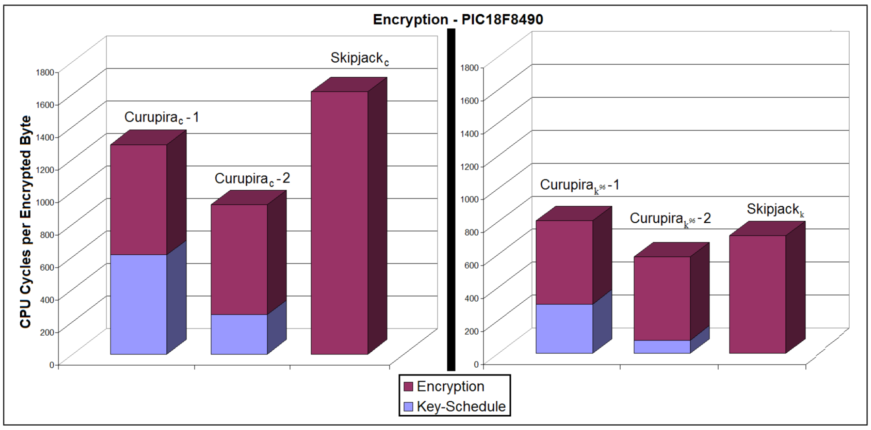


Fig. 4. Comparison between the cipher encryption speeds on the PIC18F8490

The figure shows that CURUPIRA$_c$-1 and CURUPIRA$_c$-2 are respectively $\approx$ 20% and $\approx$ 45% faster than Skipjack$_c$, with the round keys computed on demand. Despite this very positive result, it should be carefully considered since the measured number of cycles includes not only the operations directly involved in the encryption process but also a non-negligible number of auxiliary ones. The influence of these secondary operations is less expressive on both Skipjack$_k$ and CURUPIRA$_{k^{96}}$ and, as depicted in the right side of Figure 4, CURUPIRA$_{k^{96}}$-2 is still $\approx$ 18% faster than Skipjack, while CURUPIRA$_{k^{96}}$-1 is $\approx$ 12% slower. One can see that the cost of both CURUPIRA$_{k^{96}}$ versions in this figure are approximately the ones theoretically calculated in Section 4.6.

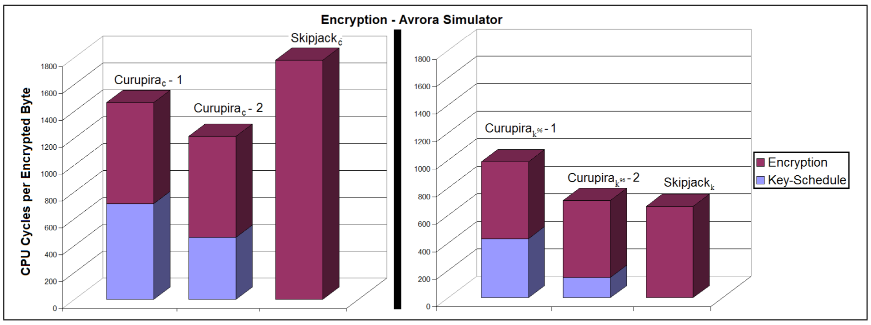


Fig. 5. Comparison between the cipher encryption speeds on the Avrora Simulator

The results on the Avrora Simulator were slightly different from those in the PIC18F8490, as depicted in Figure 5. Skipjack$_k$ speed was considerably improved by this platform change, running about 30% and 4% faster than CURUPIRA$_{k^{96}}$-1 and CURUPIRA$_{k^{96}}$-2, respectively. A further analysis of the assembly codes show that these results were caused by the influence of the compiler, which were able to apply different optimizations to each algorithm. In contrast, this unexpected behavior was not observed with CURUPIRA$_c$ and Skipjack$_c$ implementations, which sustained the relative performances presented on the PIC18F8490.

5.3 Results: 32-bits platforms

The encryption speed of each cipher on the 32-bits platform, with different key sizes (and, thus, number of rounds), is depicted in Figure 6. It is important to point out that, as all round keys are computed at cipher initialization, there is no difference between the encryption speeds of CURUPIRA-1 and CURUPIRA-2.

We obtained similar results for AES and CURUPIRA with the same number of rounds (note the additional graph entry where, for the sake of comparison, both ciphers were tested with the same number of rounds for each key size). This is an expected result since both ciphers adopt well-known optimizations for the Wide Trail Strategy family. On the other hand, the modest result of Skipjack

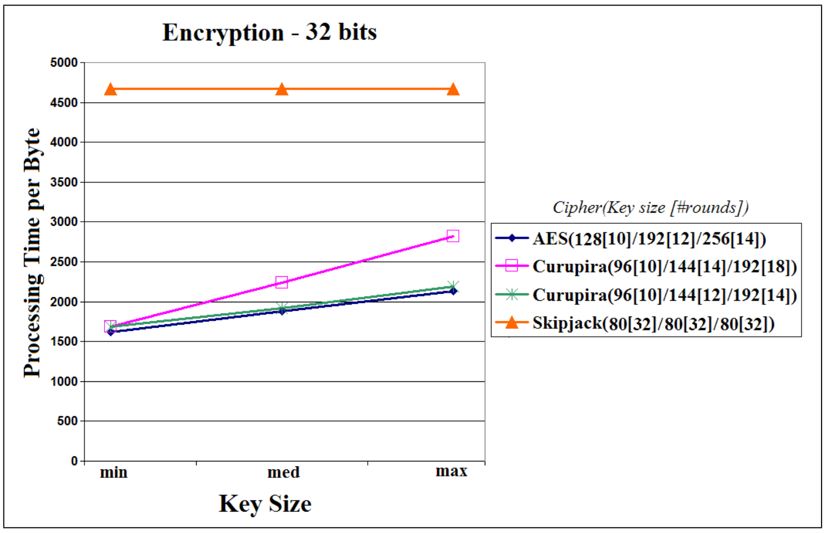


Fig. 6. Encryption Performance x Number of Rounds - 32 bits

(about 3 times slower than the other ciphers) may seem surprising at first sight, since its performance usually is the main factor for its adoption on constrained platforms. However, this can be explained by its 16-bit oriented operations, very attractive on constrained processors but less adapted to fully take advantage of the higher number of bits available on powerful platforms. Both AES and Curupira, though, can easily be implemented to operate over 32- and 24-bit words (columns), respectively.

The processing time involved on the ciphers key-schedules was also measured. As depicted in Figure 7, while Curupira and AES presented similar speeds, Skipjack was about 10 times slower. As this operation has to be performed a single time (at initialization), the impact on the cipher overall operation is reduced on scenarios where the keys are not frequently changed.

6 Conclusions

We have described a new and faster key-schedule proposal for the Curupira block cipher. As a drawback, when compared to the original specification, it has a lower level of security against related-key attacks. However, according to our security analysis, both versions of the full cipher are secure against cryptanalysis.

We also presented a benchmark comparing Curupira, Skipjack and AES in terms of performance and memory occupation, both on constrained and powerful platforms. According to the results obtained, the proposed block cipher is fast and compact, especially when using the new key-schedule presented in this paper. While Skipjack is considered a good candidate for constrained scenarios, such as LBAs dependent on sensors and low-power mobile devices, Curupira is

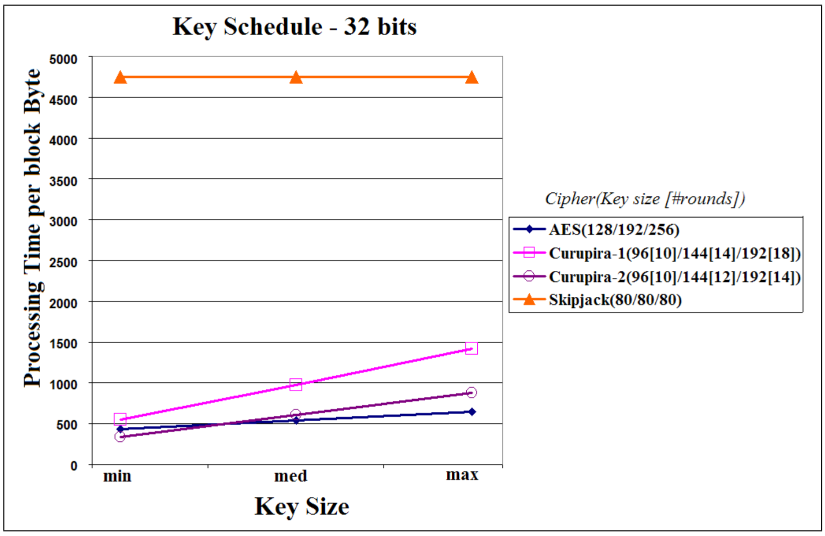


Fig. 7. Key Schedule Performance x Number of Rounds - 32 bits

a suitable alternative to increase the security level and potentially improve performance, introducing a reduced impact in terms of memory usage. Also, when more powerful platforms are also available, the several optimizations allowed by the CURUPIRA can be deployed to obtain an even higher performance in the whole network.

All together, these results show that the CURUPIRA block cipher is a useful solution for providing data encryption at low cost, being recommended for constrained-resource devices and for applications based on such platforms, such as WSNs and LBAs.

6.1 Future and Ongoing work

We are currently working on the deployment of CURUPIRA on a real WSN in order to evaluate its impact (especially in terms of energy consumption) in some significant scenarios. Also, we are developing a new MAC algorithm named MARVIN, designed to provide a low-cost authenticated-encryption scheme on WSNs, particularly when used in conjunction with CURUPIRA.

6.2 Acknowledgments

We would like to thank Richard Gold for his useful comments and the review of this paper.

References

1. 3GPP. Specification of the 3gpp confidentiality and integrity algorithms document 2: Kasumi specification. Technical report, 3GPP, 1999.

2. Atmel. *AVR 8-Bit RISC processor - ATmega8535 (90LS8535)*, 2006.
3. Atmel. *AVR 8-Bit RISC processor - ATmega128 e ATmega128L*, 2007.
4. P. Barreto. The Skipjack block cipher – 32 bit implementation. http://planeta.terra.com.br/informatica/paulobarreto/skipjack32.zip, 1998.
5. P. Barreto. The AES block cipher (rijndael) – 32 bit implementation. http://planeta.terra.com.br/informatica/paulobarreto/JEAX.zip, 2003.
6. P. Barreto and M. Simplicio. Curupira, a block cipher for constrained platforms. In *Anais do* 25^o *Simpsio Brasileiro de Redes de Computadores e Sistemas Distribudos - SBRC 2007*, volume 1, pages 61–74. SBC, 2007.
7. P. S. L. M. Barreto and V. Rijmen. The Anubis block cipher. In *First open NESSIE Workshop*, Leuven, Belgium, November 2000. NESSIE Consortium.
8. P. S. L. M. Barreto and V. Rijmen. The Khazad legacy-level block cipher. In *First open NESSIE Workshop*, Leuven, Belgium, November 2000. NESSIE Consortium.
9. E. Biham, A. Biryukov, and A. Shamir. Cryptanalysis of skipjack reduced to 31 rounds using impossible differentials. In *Advances in Cryptology – Eurocrypt'99*, volume 1592 of *Lecture Notes in Computer Science*, pages 55–64. Springer, 1999.
10. M. Ciet, G. Piret, and J. Quisquater. Related-key and slide attacks: Analysis, connections, and improvements - extended abstract. In *23rd Symposium on Information Theory in the Benelux, Louvain-la-Neuve, Belgium*, pages 315–325, 2002.
11. J. Daemen. *Cipher and hash function design strategies based on linear and differential cryptanalysis.* Doctoral dissertation, Katholiek Universiteit Leuven, March 1995.
12. J. Daemen and V. Rijmen. The block cipher BKSQ. In *Smart Card Research and Applications – CARDIS'98*, volume 1820 of *Lecture Notes in Computer Science*, pages 236–245. Springer, 1998.
13. N. Ferguson, J. Kelsey, S. Lucks, B. Schneier, M. Stay, D. Wagner, and D. Whiting. Improved cryptanalysis of Rijndael. In *Fast Software Encryption – FSE'2000*, volume 1978 of *Lecture Notes in Computer Science*, pages 213–230. Springer, 2000.
14. B. R. Gladman. AES second round implementation experience. http://fp.gladman.plus.com/cryptography technology/aesr2/index.htm, 2000.
15. G. Guimaraes, E. Souto, D. Sadok, and J. Kelner. Evaluation of security mechanisms in wireless sensor networks. In *ICW '05: Proceedings of the 2005 Systems Communications*, pages 428–433. IEEE Computer Society, 2005.
16. M. Healy, T. Newe, and E. Lewis. Efficiently securing data on a wireless sensor network. *Journal of Physics*, 76, 2007.
17. J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. Culler, and K. Pister. System architecture directions for networked sensors. In *Architectural Support for Programming Languages and Operating Systems*, pages 93–104, 2000.
18. J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. Culler, and K. Pister. System architecture directions for networked sensors. In *Architectural Support for Programming Languages and Operating Systems*, pages 93–104, 2000.
19. C. Karlof, N. Sastry, and D. Wagner. Tinysec: a link layer security architecture for wireless sensor networks. In *2nd International Conference on Embedded Networked Sensor Systems – SenSys'2004*, pages 162–175, Baltimore, USA, 2004. ACM.
20. P. Kocher, J. Jaffe, and B. Jun. Introduction to differential power analysis and related attacks. Technical report, Cryptography Research Inc., 1998.
21. Y. W. Law, J. Doumen, and P. Hartel. Survey and benchmark of block ciphers for wireless sensor networks. *ACM Transactions on Sensor Networks (TOSN)*, 2(1):65–93, 2006.

22. P. Levis and D. Culler. Maté: a tiny virtual machine for sensor networks. In *ASPLOS-X: Proceedings of the 10th international conference on Architectural support for programming languages and operating systems*, pages 85–95. ACM, 2002.
23. T. Li, H. Wu, X. Wang, and F. Bao. SenSec design. Technical report, InfoComm Security Department, February 2005.
24. D. Liu, P. Ning, and R. Li. Establishing pairwise keys in distributed sensor networks. In *CCS'03: Proceedings of the 10th ACM conference on Computer and communications security*, pages 52–61. ACM, 2003.
25. M. Luk, G. Mezzour, A. Perrig, and V. Gligor. Minisec: A secure sensor network communication architecture. In *IPSN'07: Proceedings of the 6th international conference on Information processing in sensor networks*, pages 479–488, New York, NY, USA, 2007. ACM.
26. F. J. MacWilliams and N. J. A. Sloane. *The theory of error-correcting codes*, volume 16. North-Holland Mathematical Library, 1977.
27. M. Matsui. New block encryption algorithm MISTY. In *Fast Software Encryption – FSE'97*, volume 1267 of *Lecture Notes in Computer Science*, pages 54–68. Springer, 1997.
28. A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone. *Handbook of Applied Cryptography.* CRC Press, Boca Raton, USA, 1999.
29. Microchip. *PIC18F8490 Datasheet*, 2006.
30. R. Müller, G. Alonso, and D. Kossmann. SwissQM: Next generation data processing in sensor networks. In *CIDR*, pages 1–9, 2007.
31. J. Nakahara. Analysis of Curupira block cipher. In *Anais do 8º Simpsio Brasileiro em Segurana da Informao e Sistemas Computacionais*, 2008.
32. NIST. *Federal Information Processing Standard (FIPS 197) – Advanced Encryption Standard (AES).* National Institute of Standards and Technology, November 2001. `http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf`.
33. NSA. *Skipjack and KEA Algorithm Specifications, version 2.0.* National Security Agency, May 1998.
34. R. L. Rivest. The RC5 encryption algorithm. In *Fast Software Encryption – FSE'94*, volume 1008 of *Lecture Notes in Computer Science*, pages 86–96. Springer, 1995.
35. R. Roman, C. Alcaraz, and J. Lopez. A survey of cryptographic primitives and implementations for hardware-constrained sensor network nodes. *Mobile Networks and Applications*, 12(4):231–244, 2007.
36. B. Titzer, D. Lee, and J. Palsberg. Avrora scalable simulation of sensor networks with precise timing. Center for Embedded Network Sensing Posters - Paper 93, 2004.

The name

According to a Brazilian legend, the Curupira is a spirit of nature and protector of the forests. It assumes the form of a boy with red hair, whose feet are turned backwards. This way, when hunters think they are on the right trail to get it, they in fact are going to the wrong direction, getting confused and lost. This should also work against cryptanalysts :-).

Author Index